AN EVOLUTIONARY SOFTWARE PROJECT MANAGEMENT MATURITY MODEL FOR DEVELOPING COUNTRIES

by

ANEERAV SUKHOO

submitted in accordance with the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in the subject

COMPUTER SCIENCE

at the

UNIVERSITY OF SOUTH AFRICA

PROMOTER: PROFESSOR M. M. ELOFF

JOINT PROMOTERS: PROFESSOR J. A. VAN DER POLL
PROFESSOR A. BARNARD

MARCH 2009

Abstract

The evidence of project management is known for centuries as can be observed from the construction of the pyramids of Egypt and the Great Wall of China. In the modern age, project-based approaches are increasingly being adopted in almost all areas of product development. Project management has emerged as a specialised discipline since the importance of completing projects within time, cost and quality constraints is crucial for organisations to continue to survive in the competitive world.

Project Management, in general, has been developed under the assumption of economic, political, cultural and social rationality. Little is known about indigenous project management in developing countries. The concepts and principles put forward in Western/European countries have been applied to developing countries without much prior studies. Developing nations are struggling constantly to maintain their projects within the constraints of time, cost and quality. Software development companies have found it difficult to adopt methodologies/models/standards that have shown evidence of success in the developed world.

This thesis comprises the study of software project management in developing countries. Mauritius, as an instance of developing countries, with problems related to social, economic, cultural and political conditions are discussed. However, these conditions differ from country to country. It is believed that adjustments are required in a software project management framework to fit the requirements of a country.

An evolutionary software project management maturity model is proposed for managing software development in developing countries. This model adopts an evolutionary approach, whereby areas of interest (called key process areas) progressively attain maturity. Three levels of maturity are defined along with key process areas that are applicable over all the maturity levels (called the

continuous process improvement group of KPAs). The model is also applied to two software projects in Mauritius to test its effectiveness. Given the studies carried out and its successful application to the Mauritian context, this model for software project management is expected to contribute towards a higher software project success rate. Notwithstanding the application in the Mauritian context, it is plausible that other developing countries may also customise this model as similar problems occur across these countries.

Acknowledgements

This thesis concludes a period of 7 years of research towards a PhD degree in Computer Science. Throughout this period, my promoter Professor M. M. Eloff and joint promoters Professor J. A van der Poll and Professor A. Barnard have provided invaluable help. I wish to thank them for their guidance, encouragement and motivation during the preparation of the thesis.

I wish to place on record the excellent guidance and encouragement of Professor A. Barnard who was initially the main promoter of this thesis. Unfortunately, he has not been able to continue with his assignment till the finalisation of the thesis due to ill health.

I would also like to thank Ms. M. Malan, the librarian, for her help in searching for articles related to the research and sending me same either by email or by post. In addition, during my visits to UNISA, she has been very kind to help me to access various publications in the library.

I wish also to thank all those who have participated in surveys conducted in Mauritius. An attempt for the first survey was done in several developing countries, but due to the low response rate the research was restricted to Mauritius.

Declaration

The thesis entitled "An Evolutionary Software Project Management Maturity Model for Developing Countries" is based on work carried out by the author between April 2002 and November 2008.

All the work and ideas in this thesis are original unless otherwise acknowledged either in the text, or by reference.

This work has not been submitted for any degree at this or any other university.

The views expressed in this thesis are those of the author and has been vetted by the promoter and joint promoters.

Table of Contents

Chapter 8: Continuous process improvement

List of Figures

List of Tables

Acronymns

CMMI	Capability Maturity Model Integration
CPM	Critical Path Method
$ESPM^3$	Evolutionary Software Project Management Maturity Model
ICT	Information and Communication Technologies
IS	Information System
ISO	International Standards Organisation
IT	Information Technology
KPA	Key Process Area
PDCA	Plan-Do-Check-Act
PERT	Project Evaluation and Review Technique
PM	Project Management
PMBOK®	Project Management Body of Knowledge
PRINCE 2®	Projects IN Controlled Environments
RFP	Request for Proposal
SEI	Software Engineering Institute
SPM	Software Project Management
TQM	Total Quality Management
UK	United Kingdom
US	United States of America
WBS	Work Breakdown Structure

Published papers

The following papers have been published during the research reported on in this thesis:

Sukhoo, A.; Barnard, A.; Eloff, M.M. and Van der Poll, J.A. 2004. Towards a framework for evaluating strengths and weaknesses of software projects. *Proceedings of the 2004 PMSA Global Knowledge Conference*. Johannesburg. South Africa. 193-199.

Sukhoo, A.; Barnard, A.; Eloff, M.M. and Van der Poll, J.A. 2004. A survey of project management tools, techniques and methodologies used in Mauritius: The current status. *Proceedings of the 2004 PMSA Global Knowledge Conference*. Johannesburg. South Africa. 320-329.

Sukhoo, A.; Barnard, A.; Eloff, M.M. and Van der Poll, J.A. 2005. An assessment of Software project maturity in Mauritius. *Informing Science Institute journal*. Arizona. USA. 671-690. (This paper was fast tracked to the Informing Science Institute journal).
(***This paper was also reprinted in Projects & Profits. July 2006. The ICFAI University Press. Hyderabad. India***).

Sukhoo, A.; Barnard, A.; Eloff, M.M.; Van der Poll, J.A. and M. Motah. 2005. Accomodating soft skills in software project management. *Informing Science Institute journal*. Arizona. USA. 691-703. (This paper was fast tracked to the Informing Science Institute journal).

Sukhoo, A.; Barnard, A.; Eloff, M.M. and Van der Poll, J.A. 2007. An evolutionary software project management maturity model for Mauritius. *Interdisciplinary Journal of Information, Knowledge, and Management*. Slovenia. 99-118. (This paper was fast tracked to the Interdisciplinary Journal of Information, Knowledge, and Management journal).

Organisation of the Thesis

This thesis comprises four parts as follows:

Part 1: Introduction and Background.

In part 1, the introduction and background of the thesis are presented. The aim of the research, problem statement, objectives and deliverables are provided. The research approach is also discussed.

Part 2: Literature survey and status of Software Project Management.

In part 2 of the thesis, the literature survey is discussed based on the principles of software and generic project management methodologies/models/standards available and in use. The status and problems in the area of software project management in developed countries (like the United Kingdom and United States) and African developing countries are also exposed. The status of software project management in Mauritius as a developing country is also presented.

Part 3: Proposed maturity model.

In part 3, a maturity-based software project management model with three maturity levels is proposed for developing countries. This model called the Evolutionary Software Project Management Maturity Model (ESPM^3) is developed based on surveys and interviews carried out in Mauritius. Two case studies are also presented to show the applicability of the ESPM^3 in Mauritius as a developing country.

Part 4: Conclusion and future work.

In this part, the summary, conclusion and future work that may be undertaken are discussed.

Part 1

Introduction and Background

Chapter 1

Introduction and Background

Table of contents

1.1 Introduction

In this chapter, an overview of the research is presented including the aim, problem statement, objectives and deliverables. In addition, the research approach is elaborated on and finally the chapter layout of the thesis is given. In the thesis, research is conducted and a software project management maturity model is proposed for developing countries. However, Mauritius is chosen as an instance of developing countries. It is noted that all the project management concepts of Western origin are not universally valid (Muriithi and Crawford, 2003; Turner, 1993). Some of the concepts put forward in developed countries may not be totally applicable to the developing world.

There is also a general perception that traditional engineering projects (i.e. projects that are not of a software nature) are better managed than Information Technology (IT) projects (Baber, 1987). According to Mullaly (2001), who agrees that there is no difference in the management of the two types of projects (traditional engineering and software projects), the greatest cost of a construction project is the materials used while the greatest point of variability in an IT project lies in its human resources. Brooks (1995) as well as Hughes and Cotterell (2006) belong to another school of thought viewing software project management (SPM) as being different from that of managing traditional engineering projects.

It is noted that the two different types of projects mentioned above are defined in much the same way (Sukhoo, Barnard, Eloff and Van der Poll, 2004a), for instance, one definition of a project is (Gray and Larson, 2000):

> *A project is a complex, non-routine effort limited by time, budget, resources and performance specifications designed to meet customer needs.*

Three project parameters, namely time, cost and quality (performance) can be identified from the above definition. A successfully managed project is considered to be one that is completed on or before deadline (Time parameter), within budget (Cost parameter) and at a specified level of quality (Quality parameter). A project normally has a finite life span, starting at its conception till its close out.

Various organisations are increasingly relying on project management to help in project delivery on time, within budget and with respect to an agreed quality level. Software project management is no exception as this field has witnessed rapid developments during the past decades (see figure 2.2 in chapter 2; Nicholas, 2001). Specific tools and techniques have appeared as well as at least one software specific methodology/model/standard, namely the Capability Maturity Model Integration (CMMI) aimed at providing guidance that encourages process improvement in organisation (CMMI Product Team, 2002). In the quest for improvement in the management of software projects in Mauritius as a developing country, we analyse in this thesis the problems encountered and propose an appropriate software project management methodology/model/standard.

1.1.1 About the title of this thesis

The title of this thesis reflects that the work reported on in the chapters that follow is aimed at improving software project management practices and techniques for countries during their stages of development. Through this, we have concentrated on one such instance of a developing country, namely, Mauritius, a country on the south-east coast of the African continent. In this regard, Mauritius is classified as a developing country in a similar manner as is done for African countries (Macky, 2008). Hence the software project management maturity model developed in this work is customised for the Mauritian environment and it is hoped that it may in future be extended to facilitate SPM practices in other developing countries as well.

1.1.2 Software project management in Mauritius

The software industry in Mauritius is growing rapidly and the Mauritian Government has decided to develop the Information and Communication

Technologies (ICT) sector as the fifth pillar of the economy (Government portal, 2006) besides sugar, textile, tourism and the financial services sector. Many foreign companies like Infosys (US Department of State, 2007) and Accenture (2006), among others, have opened branches in Mauritius to seize opportunities offered by the country in terms of building infrastructure, telecommunications and commitment of motivated political leaders. These companies have expertise in managing their projects. However, it is uncertain whether Mauritian companies can successfully emerge in the sector by managing software development in a similar manner. Popular methodologies/models/standards like the Project Management Body of Knowledge - PMBOK® (PMBOK Guide®, 2006), Capability Maturity Model Integration - CMMI (CMMI Product Team, 2002) and PRINCE 2® (CCTA, 1997) have not readily been accepted by Mauritian software development companies. It has even been observed that there is a lack of formal software project management methodologies in use by Mauritian companies (Sukhoo, Barnard, Eloff and Van der Poll, 2004b).

1.2 Background

The concept of modern project management emerged as a discipline since the time it was developed for managing the US space programme in the early 1960s (Haynes, 1989). The practice of project management was adopted by government, military services, industry, society and the information technology sector.

A plethora of generic and specific methodologies/models/standards is now available and most of these methodologies are subject to continuous improvement. Therefore, regularly updated versions are released. Project management methodologies/models/standards like PMBOK®, PRINCE 2® and the Software Engineering Institute's (SEI's) Capability Maturity Model are now widely used.

Most project management methodologies/models/standards are generic in nature, that is, they cut across various disciplines and they are applied globally. Some studies (Muriithi and Crawford, 2003; Turner, 1993) revealed that project management concepts are not universally applicable because economic rationalities are too often assumed and social, cultural, political, and legal variations occur across nations but these are often ignored. In addition, developing countries constantly face problems of insufficient skilled staff, funds, and political and social incentives. Some problems, amongst others, that are evident in developing countries are (Muriithi and Crawford, 2003):

(i) lack of staff prevents development of extensive documentation

(ii) lack of funds is a hurdle towards acquisition, training and consultancy in respect of adopting a software project management methodology/model/standard.

(iii) adopting a methodology of Western origin with insufficient staff may lead to pressure on the part of software development staff to meet targets while ignoring social, cultural and community development activities.

Coupled with problems of developing countries, projects have revealed factors like unclear objectives, poor planning, new technology and a lack of project management methodology (Jalote, 2002).

The advantage for Mauritius, although belonging to the developing world, is that it is aspiring to become the Silicon Valley of the African region and the Mauritian Government has set a vision to transform the island into a Cyber Island based on the Singaporean or Taiwanese model (Eid, 2002). The development of Mauritius into a Cyber Island favours a breeding ground for ICT projects ranging from short-term through medium-term to long-term implementations. The ICT sector is expected to emerge as the fifth pillar of the economy besides sugar, textile, tourism and the financial services sector (Government portal, 2006).

1.3 Aim of the research

Western software project management methodologies are being applied in developing countries in spite of the differences in economies, cultures, politics and legal frameworks. The situation of Mauritius has been analysed through surveys conducted in Mauritius (Sukhoo et al., 2004b; Sukhoo, Barnard, Eloff and Van der Poll, 2005a).

Against the background sketched in section 1.2, in this thesis we analyse the various problems that are specific to software development in Mauritius and put forward an appropriate software project management methodology/model/standard. In addition, recommendations for relevant existing tools and techniques are also proposed.

1.4 Problem Statement, objectives and deliverables

According to Muriithi and Crawford (2003), software project management methodologies of Western origin are not universally applicable and, therefore, developing countries have to be cautious in the selection of an appropriate methodology/model/standard, tools and techniques for the management of their software projects. The case of Mauritius, as an instance of developing countries, is considered for the purpose of this study.

The main problem area to be addressed in this thesis is:

Software project management methodologies/models/standards of Western origin are not universally applicable and developing countries have specific problems that need to be addressed so as to enhance the success rate of projects.

This problem statement can be disseminated into the following **primary objective**:

To develop a software project management methodology/model that can enhance the success rate of software projects in developing countries.

A solution to the problem described by the above problem statement may be achieved by the formulation of an appropriate software project management framework by taking cognisance of the problems specific to developing countries as mentioned in section 1.2.

The following **secondary objectives** with their corresponding **deliverables** are proposed:

1.4.1 Objective 1 and first deliverable

Mauritius has embarked on an ambitious initiative to develop ICT as the fifth pillar of the economy (Government portal, 2006) in order to transform the country into a Cyber Island. The software industry is inevitably one of the sectors that has to contribute to the realisation of the vision of the Government. Mauritius is a developing country and research conducted by Muriithi and Crawford (2003) demonstrated a need for a framework for project management with an indigenous approach. Therefore, it is required to identify possible shortcomings in available project management methodologies/models/standards in order to come up with a software project management methodology/model/standard for developing countries.

Objective 1: To **identify shortcomings** in available project management methodologies/models/standards and to develop a software project management methodology/model/standard for developing countries. The methodology/model/standard will be the **first** deliverable.

1.4.2 Objective 2 and second deliverable

Several tools and techniques are available to help in the management of projects, e.g. Gantt charts, function points, object points and estimation by analogy and expert judgement. The appropriate tools have to be identified for use along with the software project management methodology/model/standard.

Objective 2: To **identify tools and techniques** to be used with the software project management methodology/model/standard. These tools and techniques will be the **second** deliverable.

A range of project management tools and techniques are available. The tools and techniques commonly used in Mauritius (as an instance of developing countries) will be identified.

1.5 Research Approach

Further to two research studies carried out (the full papers are found in the section of published papers), namely "Towards a framework for evaluating strengths and weaknesses of software projects" (Sukhoo et al., 2004a) and "A survey of project management tools, techniques and methodologies used in Mauritius: The current status" (Sukhoo et al., 2004b), the need for a different software project management methodology for Mauritius was made clear. The rapid development of the ICT sector and the specificities of Mauritius as a developing country necessitate a practical project management approach. This thesis addresses specifically the software development sector as it is noted that software projects should be treated differently as compared to traditional engineering projects (Sukhoo et al., 2004a; Hughes and Cotterell, 2006).

The survey carried out by Sukhoo et al. (2004b) revealed that many software development companies in Mauritius do not use any formal project management methodology. In order to encourage these companies in adopting a suitable methodology, it is important to strike a balance between using an extensive and bureaucratic methodology/model/standard on the one hand and providing a practical approach for managing software projects on the other.

Another survey was also conducted by Sukhoo et al. (2005a) to assess the maturity level of software development organisations in Mauritius. In addition, a combined survey and interviews were conducted with eleven pertinent companies in Mauritius to gather in-depth information on the issues to be considered for the development of a software project management methodology/model/standard (Sukhoo, Barnard, Eloff and Van der Poll, 2007).

On the strength of these findings, a new model was proposed. The model was applied to two companies as case studies to assess its applicability.

Hence, the research approach comprised:

(i) An extensive literature study of the various project management methodologies/models/standards with particular emphasis on software project management (presented in chapters 2 and 5).

(ii) An extensive literature study of problems encountered in the software project management field in the US, UK and African developing countries (presented in chapter 3).

(iii) Surveys conducted in Mauritius to assess the status of software project management (presented in chapters 4 and 6).

(iv) The development of a software project management maturity model that was tested in two environments to prove its applicability.

1.6 Thesis layout

Figure 1.1 serves as a roadmap for the thesis.

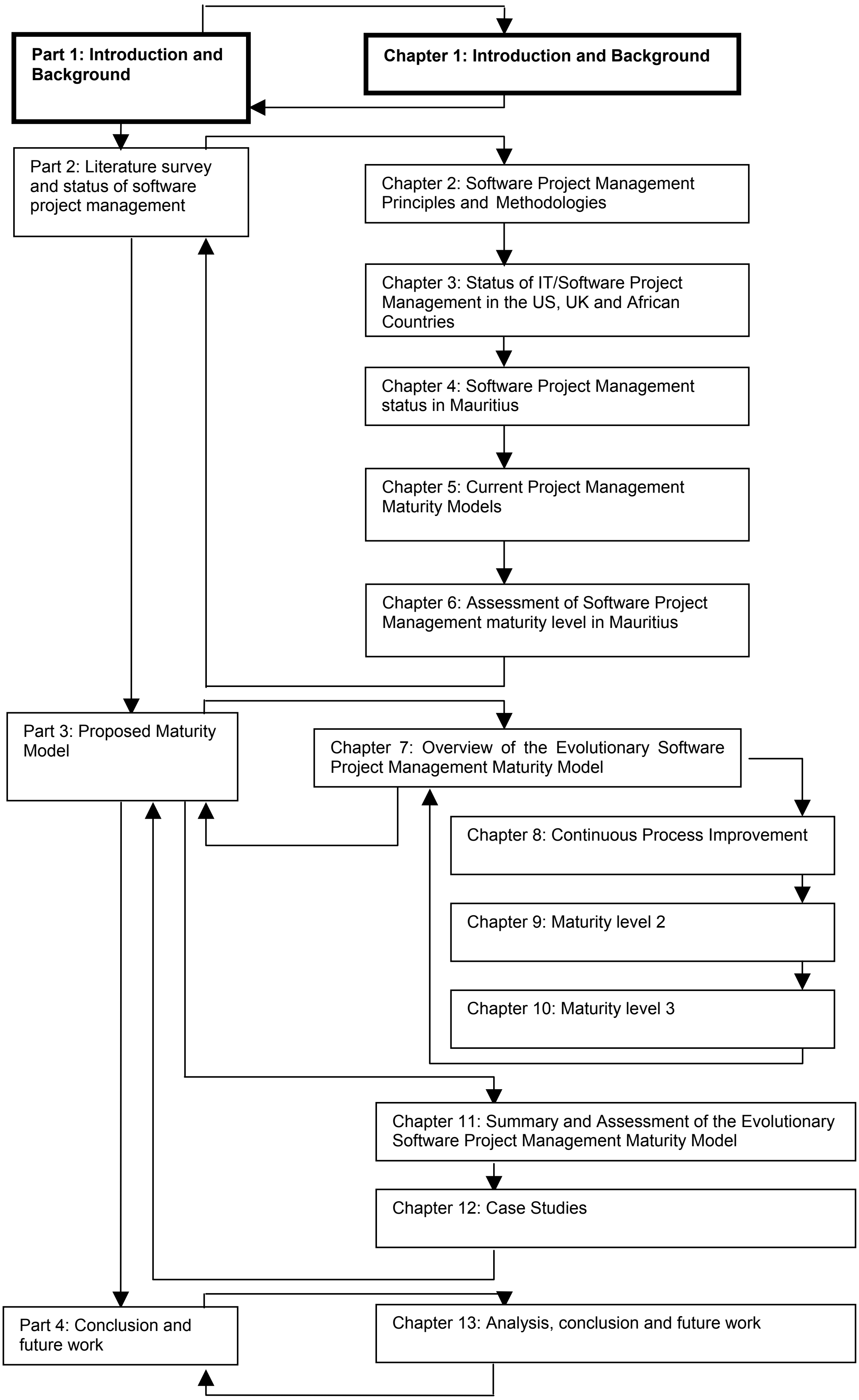


Figure 1.1: Chapter 1 in context within the overall thesis

Part 1: Introduction and Background

Part 1 comprises ***chapter 1***, in which an introduction to the research is presented. The motivation, approach, goal and objectives of the research are also described in this part of the thesis.

Part 2: Literature survey and status of software project management

In Part 2, made up of chapters 2, 3, 4, 5 and 6, the development in the field of project management in general is investigated and the focus is provided for software project management in particular. The problems faced by Mauritius in the field of software development are also covered and the motivation for an appropriate software project management model for the country is presented.

In ***chapter 2***, software project management principles and methodologies/models/standards are presented. An introduction to software project management is provided as well as a brief history of developments in the discipline. The state of software project management in developed countries is discussed and an overview of several methodologies/models/standards successfully used in these countries is given.

In ***chapter 3***, the status of project management in the US, UK and African developing countries is discussed with particular emphasis on software project management. Several studies carried out in Africa are analysed with a view to investigate the problems encountered and assess the validity of applying software project management concepts, tools and techniques of Western/European origin.

In ***chapter 4***, a survey on the status of software project management in Mauritius is presented. The tools, techniques and methodologies used in Mauritius are assessed and the problems specific to the country are highlighted.

A literature survey on maturity models is presented in ***chapter 5***. As it has been established that there is a correlation between the maturity level of an organisation and project success (Schiltz, 2003; Sonnekus and Labuschagne, 2004), the study is considered to be of invaluable importance in order to come up with a software project management methodology/model/standard suitable for Mauritius.

Further to a study of maturity models and the insights gathered, a study of the maturity level of software organisations in Mauritius was carried out. The results of the study are presented in ***chapter 6***.

Part 3: Proposed maturity model

Part 3 comprises chapters 7, 8, 9, 10, 11 and 12 in which the proposed software project management maturity model is described. Different maturity levels are described and summarised. Finally, the application of the model in the Mauritian context is discussed.

In ***chapter 7***, an overview of the proposed maturity model, called the Evolutionary Software Project Management Maturity Model, $ESPM^3$, is provided. The model is explained in terms of maturity levels, each comprising key process areas that are focus areas (or areas of interest) aimed at achieving a particular maturity level. A graphical representation of the model is given and a mathematical relationship between maturity levels and key process areas is established.

In ***chapter 8***, emphasis is laid on the continuous process improvement key process areas. These key process areas are considered at all maturity levels. The areas covered are soft skills management, change management, software specific focus, Integration management and environmental management.

In ***chapter 9***, maturity level 2 is described as the basic project management level. Three key process areas, namely time, cost and quality management are described. The tools and techniques employed are also presented.

In ***chapter 10***, maturity level 3 is described as the organisational project management level. Here, three key process areas are identified, namely human resource management, risk management and contract management. This level assumes the pre-requisites of all key process areas of level 2.

In ***chapter 11***, a summary of the Evolutionary Software Project Management Maturity Model (ESPM^3) is presented. This summary can provide a project manager, knowledgeable in ESPM^3, with a quick guideline for the application of the model.

In ***chapter 12***, the application of the ESPM^3 to two software projects in Mauritius is discussed.

Part 4: Conclusion and future work

In this part of the thesis, comprising ***chapter 13***, an analysis and conclusion of the research is provided. In addition, the directions for future work are also indicated.

1.7 Conclusion

In this chapter, the introduction, background, aim, objectives, deliverables and research approach for this thesis were discussed. The chapters grouped into four parts were also briefly presented.

The scene was set for the development of a software project management maturity model for developing countries by considering Mauritius as an instance. In this thesis, the proposed software project management model is tested in two companies to show its applicability. In order to pursue further research, the plausibility of future work is also indicated.

In the next chapter, the principles of software project management and available project management (generic as well as software specific) methodologies/models/standards are discussed as part of the literature study.

The rest of the literature study as well as the status of software project management are presented in the remaining four chapters of part 2 of the thesis.

Part 2

Literature survey and status of Software Project Management

Chapter 2

Software Project Management Principles and Methodologies/Models/Standards

---

Table of contents

2.0 Chapter layout

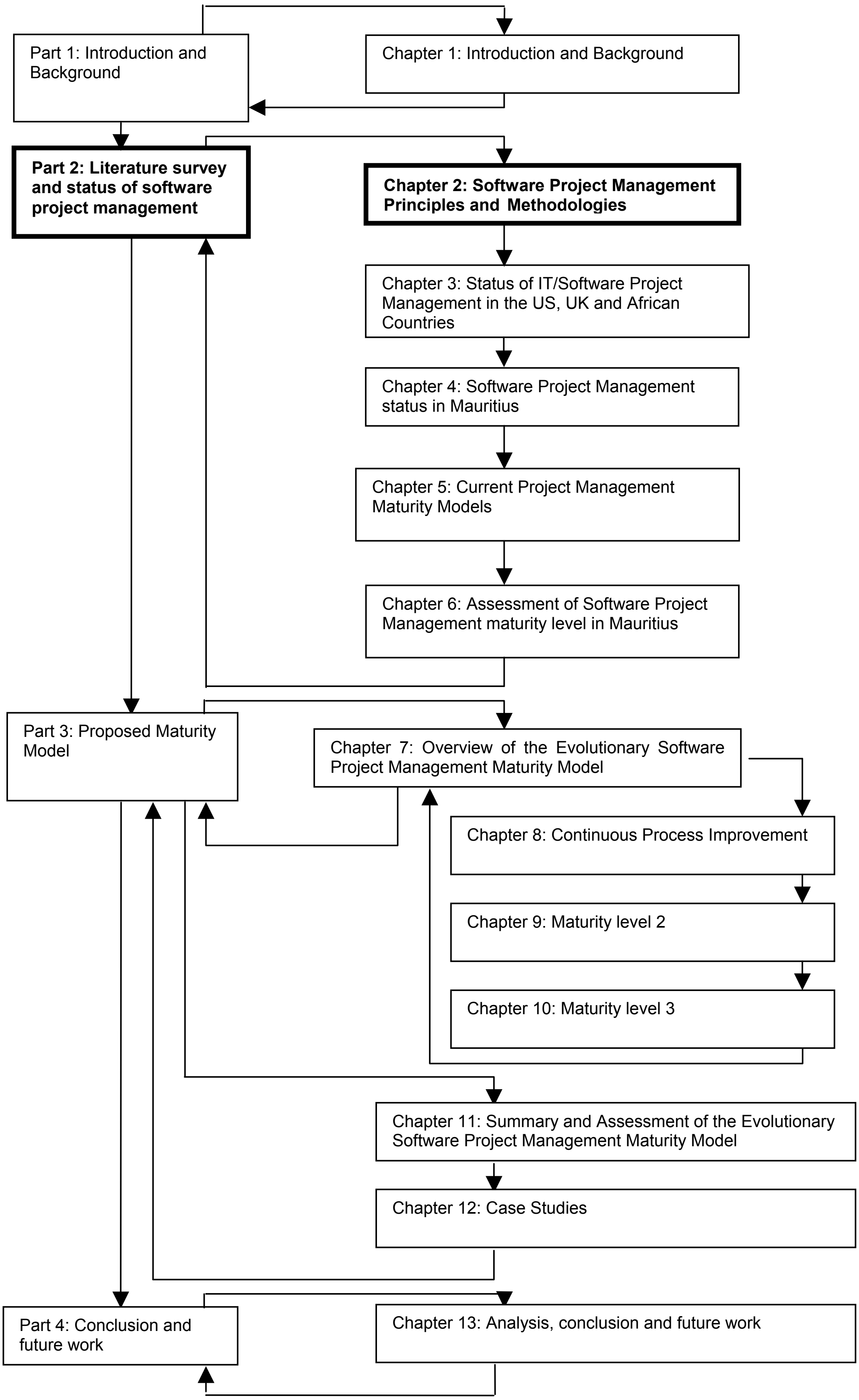


Figure 2.1: Chapter 2 in context within the overall thesis

2.1 Introduction

A brief exposé of the various chapters of the thesis was presented in the previous chapter. In the current chapter, project management in general is introduced with a brief history on its evolution. Various projects, like the pyramids of Egypt and the Great Wall of China (Nicholas, 2001), are believed to have used some form of project management techniques. Some projects accomplished in ancient civilizations are now considered to be wonders of the world. Modern project management is regarded as originating around the beginning of the 20th century. Various tools and techniques that have been developed are discussed in this chapter and some major successful projects are also highlighted. In this chapter, the principles of software project management are discussed in terms of the tools and techniques that are commonly used as well as the different phases through which software projects are managed. In supporting these principles, a number of tools and techniques, for example Gantt charts, PERT diagrams, COCOMO model, function point analysis, object point analysis and a few software development process models are considered. The reader is also exposed to a plethora of project management methodologies/models/standards in use.

2.1.1 Aim and structure of this chapter

The aim of this chapter is to introduce software project management principles and methodologies/models/standards by starting with a general discussion on project management. Having taken note of the definition of a project in section 1.1 of chapter 1, this chapter begins with a generic definition of traditional project management followed by a definition of software project management. According to the PMBOK® Guide (2006):

Project management is the application of knowledge, skills, tools and techniques in order to meet or exceed stakeholder requirements from a project. *…………....Definition 2.1*

Generic project management tools and techniques have been used for a long time for all types of projects (McDonald, 2001). Various project management methodologies/models/standards like PMBOK®, PRINCE 2® and APMBOK have been developed and used to manage projects. However, traditional projects were found to be different from software projects in terms of visibility, complexity and flexibility (Sukhoo et al., 2004a). Hughes and Cotterell (2006) together with Brooks (1995) also agreed about the unique characteristics of the software development sector. Taking into account the difference between traditional engineering and software projects, Nienaber and Cloete (2003) defined software project management as:

Software Project Management (SPM) involves the management of all aspects and issues that are involved in the development of a software project, namely scope and objective identification, planning, evaluation, project development approaches, software effort and cost estimation, activity planning, monitoring and control, risk management, resource allocation and control, as well as managing contracts, teams of people and quality. *…………...Definition 2.2*

The principles of SPM would cover all aspects like scope, objective identification, planning, etc. as listed in definition 2.2.

The remainder of the chapter is structured as follows:

Section 2.2 – In this section, a literature overview of the evolution of project management mainly since the 1900s is presented. The development of project management tools such as the Gantt chart (Gido and Clements, 1999) and PERT (Gray and Larson, 2000) and their application to successful projects are discussed. It can also be noted that the principles of traditional project management have paved their way to various types of projects including IT and software projects.

Section 2.3 – In this section, the principles of project management are discussed. The different phases involved in project management, including software project management, are also described. The tools and techniques used as well as the need to manage resources and contracts are also highlighted in section 2.3.

Section 2.4 - The project management community has made much effort to improve the delivery of successful projects. This is clear from the plethora of project management methodologies/models/standards available. In this section, some methodologies (e.g. PMBOK® and PRINCE 2®) and maturity models (e.g. OPM3™ and CMMI) are presented. Some popular project management methodologies/models/standards, like Project Management Body of Knowledge (PMBOK®), Euromethod, BS6079:1996, ISO 12207 and PRINCE 2®, are explained in this chapter and the details are exposed in appendix A.

Maturity models are also briefly discussed in section 2.4 and the details are deferred to chapter 5 prior to the introduction of the proposed maturity model for software project management in Mauritius.

Section 2.5 – In the concluding remarks, the contents of this chapter are summarised and the major projects that marked the history of project management are highlighted. The principles of project management are deemed to be an important starting point for any further development in the field. The shortcomings of the available methodologies/models/standards and models are also considered to be important issues to address in order to increase the rate of successful projects.

2.2 History of Project Management

The construction of the pyramids in Egypt (2,550 B.C. in the case of the Great Pyramid of Giza) and the Great Wall of China (221 - 206 B.C.) show evidence of the practice of project management since a very long time (Nicholas, 2001). These magnificent works drawn from ancient history are known to have been completed at an enormous cost in human resources and time (Cleland, 1999).

However, the origin of modern project management is believed to be between the 1900s and 1950s (Carayannis, Kwak, and Anbari, 2003). This era has been particularly marked by the development of tools and techniques like the Gantt chart developed by Henry Gantt in the early 1900s (Gido and Clements, 1999) as well as the concept of the Work Breakdown Structure (WBS) which was developed with the Program Evaluation and Review Technique (PERT) by the United States Department of Defense in 1958 (Nicholas, 2001). Some major modern day projects are briefly described below.

- The construction of the Hoover dam (1931 – 1936), which was built in the Black Canyon of the Colorado River in the US with no infrastructure, mobilised around 5200 workers and was eventually constructed ahead of schedule and even under budget and it is viewed as one of the highest gravity dams in the US (Carayannis, Kwak and Anbari, 2003). The usefulness of the Gantt chart was illustrated with this project (Kozak-Holland, 2007). Since the dam was built by a consortium of six independent companies, a detailed project plan had to be prepared and the different tasks had to be well controlled and coordinated to avoid losing control of the project.

- Another project that captured much attention was the Manhattan project (1942 - 1945) that concerned the development of the first atomic bomb (Cleland, 1999). In an attempt to defend the US during World War II, the Manhattan project was initiated. Resources amounting to a 125,000 labour force and a budget of around $2 million were pooled for the development of the atomic bomb. The success of the project, although sadly devastating to Japan, is credited with the saving of lives of thousands of American personnel (Cleland, 2004).

- The Program Evaluation Review Technique (PERT) was developed in 1958 for the Polaris project (Gray and Larson, 2000) to produce nuclear missiles to be carried by submarines. The $11 billion project was successful with the launch of its first missile in 1961 (Carayannis, Kwak, and Anbari, 2003).

- The Apollo project, which aimed at exploring the moon, comprised six missions between 1969 and 1972 (Carayannis, Kwak and Anbari, 2003). PERT was used to maintain and schedule the Apollo missions. Amongst the various tasks undertaken to ensure the success of the project by the National Aeronautics and Space Administration (NASA), procurement and contracting management as well as performance management were treated as important functions. The project's success is well known for its "small step for a man and a giant leap for mankind". Project management played an important part in bringing about the success of putting a human on the moon (Cleland, 1999).

- The Internet project was initiated in 1969 by the US Department of Defense (Fitzerald and Dennis, 2002). The initial objective was to link a set of computers operated by several universities doing military research. Many organisations were later connected to the network and currently the business community is reaping enormous benefits. The Internet has grown into a global network that has made people perceive the world to be smaller given the speed at which communication is maintained.

During the period 1980 - 1994, a revolution in IT was noted with the shift from mainframe-based systems to multitasking personal computers and finally including technologies like Local Area Networks and the Internet (Carayannis, Kwak and Anbari, 2003). Project management software was readily available for PCs and therefore project management techniques were easily accessible and more flexible to use. Primavera Systems released the DOS version of Primavera project planner in 1983 (Primavera Systems, 1995) followed by Microsoft with the release of Microsoft Project 1.0 version in 1990 (Microsoft, 2007).

The Year 2000 (Y2K) bug was once a major issue worldwide. It was a man-made problem that started back in the 1950s (Carayannis, Kwak and Anbari, 2003) and was not foreseen at that time. Several government agencies had to initiate awareness programmes since 1996. Worldwide efforts and funds were poured into the replacement hardware and software as well as the modification of software. Earlier recognition of the severity of the problem would have resulted in significant savings and business risks would have been avoided (Gray and Larson, 2000). Organisations in the world made use of project management practices, tools and techniques while modifying or replacing their systems so that they were Y2K compliant before the turn of the 21st century (Carayannis, Kwak and Anbari, 2003). The Y2K project portrayed the importance of risk management to organisations throughout the world.

From the brief history of project management sketched above, the field of project management is found to be marked by the execution of important projects as well as the development of useful tools and techniques. Modern project management is believed to have started during the first half of the 20th century (Kozak-Holland, 2007) and since that period, major developments have occurred as described above.

The above developments may be represented by the following timeline:

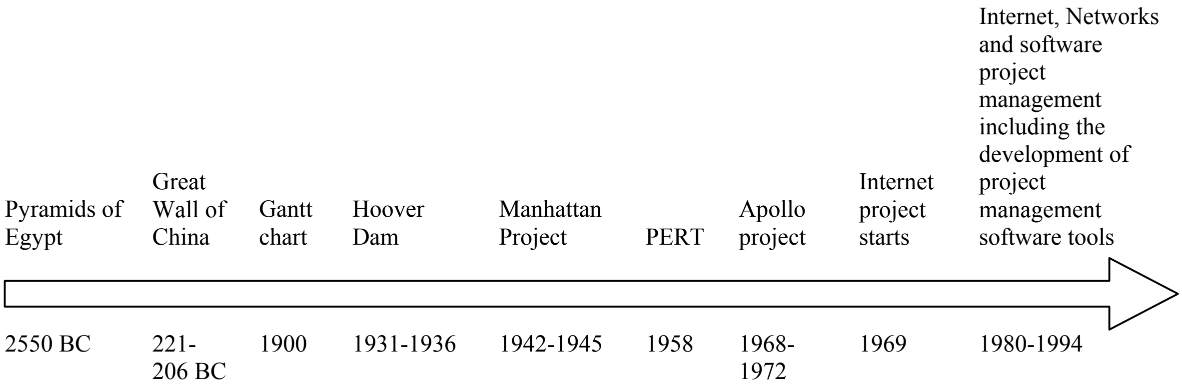


Figure 2.2: Timeline showing a partial history of Project Management

It is noted from figure 2.2 that software project management started between 1980 and 1994.

In the next section, some principles of software project management are discussed.

2.3 Software project management principles

Project management principles were developed and applied to generic projects as well as software projects. A range of methodologies/models/standards covering initiation, planning, execution, control and closing aspects (appendix A) appeared on the market. Tools and techniques as mentioned in section 2.2 were also used with success. As software tools became available, they acquired widespread popularity in project management (White and Fortune, 2002).

The scope of the project, comprising all the processes required for the delivery of the product, is defined and the objectives are agreed upon.

Four main phases can be identified during the development of a software product and they are the initiation phase, planning phase, execution phase and project closure phase (PMBOK® Guide, 2006). Similar phases are also adopted in other project management methodologies/models/standards as pointed out by Rehman and Hussain (2007) as well as Kozak-Holland (2007).

According to Haynes (1989):

- During the *initiation* phase, agreement is reached as to whether or not there is sufficient ground to proceed with a project. Commitment is secured on resources (financial as well as human resources) to be made available in order to complete the project. A documented agreement called a project charter is produced.

- The various steps involved are then planned (the *planning* phase) carefully and the plan forms the basis of the project. The use of the

Gantt chart, PERT, Work Breakdown Structure and other tools and techniques help considerably in the planning process.

- The plan is then executed (the *execution* phase) by making use of the resources available to ensure successful delivery of the output of the project (within time and cost constraints as well as meeting the user and management expectations in terms of scope and quality). The monitoring and control activities ensure that the plan is revised as the project unfolds.

- Finally, a *closing* phase marks a controlled closure of the project. The closing phase could also be added to definition 2.2

Software development effort estimation techniques were developed to better estimate the cost of software. In this context, techniques like COCOMO, Object Point and Function Point analysis (Schach, 2002; Hughes and Cotterell, 2006) appeared. Various project management models like the Waterfall model, the Spiral model, the V-process model, Rapid Application Development and Joint Application Development were also developed (Schach, 2002; O'Leary and O'Leary, 2000).

The management of resources and contracts to ensure that projects are delivered within time, cost and quality constraints are also given necessary consideration. Project management methodologies/models/standards are available to address most, if not all of these aspects.

In the next section, the methodologies/models/standards used to manage projects in general as well as software specific projects are discussed.

2.4 Available project management methodologies/models/standards

As from the 1980s, the software industry experienced a rapid boost with development of the personal computers. Software project management subsequently emerged as a promising area. As a result, the PROMPT II project management methodology appeared in 1983 (CCTA, 1997) and was originally used for IT Projects (University of Portsmouth, 2002). In the same year, the Project Management Institute (PMI) released the Project Management Body of Knowledge (PMBOK®) for managing all types of projects. Since November 1986, the Software Engineering Institute (SEI) started with the development of a process maturity framework and the initial release of the Capability Maturity Model (CMM) was available in 1992 to improve the software process (Paulk, Curtis, Chrissis and Weber, 1993). Further developments resulted in the release of Euromethod in 1996 for information systems, BS 6079:1996 for all types of projects (Hughes and Cotterell, 2006) and ISO 12207 for software development projects (Moore, 2002).

From the above historical information, it is noted that the early 1980s marked the birth of IT/Software project management with the development of the PROMPT II methodology followed by the CMM, although the concepts of project management predate the use of computers by many years (Tatnall and Shackleton, 1995). Project management tools (including Gantt charts, PERT diagrams, CPMs and WBSs) and techniques (example cost benefit analysis), have long been used before the advent of project management methodologies/models/standards. This is indicated in figure 2.2.

In the discussion that follows, some of the popular project management methodologies and models are considered. According to Rehman and Hussain (2007),

> A **methodology** provides a strategic level plan for managing and controlling an IT project. It is a combination of interrelated processes which tells us "what should be done?" but not how it has to be done?…it can be perceived as a template for initiating, planning and developing/implementing any project.

According to CMMI Product Team (2002):

> A **model** is a simplified representation of the world.

According to CMCS (2009):

> A **standard** is "a collection of knowledge areas that are generally accepted as best practice in the industry".

Currently, a plethora of project management methodologies/models/standards is available and some of them are generic in nature while others are specifically used for software project management. The following sub-sections provide an overview of methodologies that are commonly used. It is to be noted that generic methodologies are very often used for software project management as well. So, generic as well as software specific methodologies/models/standards are discussed below since all of them are being used to manage software projects. Software projects are managed by project management methodologies/models/standards that are based on sequential or maturity based models and methodologies/standards.

2.4.1 Project management methodologies/models/standards

The methodologies/models/standards discussed in this subsection are the traditional ones that follow a sequential order of execution of processes. Some examples of these models are the Project Management Body of Knowledge (PMBOK®), Euromethod, BS 6079:1996, ISO 12207 and PRINCE 2® and these are briefly discussed below. More details are found in appendix A.

2.4.1.1 Project Management Body of Knowledge PMBOK®

The first publication of PMBOK® by the Project Management Institute (PMI) dates back to 1983. The 1996 release with over 150 pages provided a realistic and mature guide to project management. A worldwide audience resulted from the free availability of the 1996 release on the PMI's website. Over the years, the number of members increased considerably. The current release is version 4 of the 2008 release.

PMBOK® is organised into nine knowledge areas, grouped into four core functions (scope, time, cost and quality) and four facilitating functions (human resource, communication, risk and procurement) with integration management tying them together (Sonnekus and Labuschagne, 2004) as shown diagrammatically in figure 2.3.

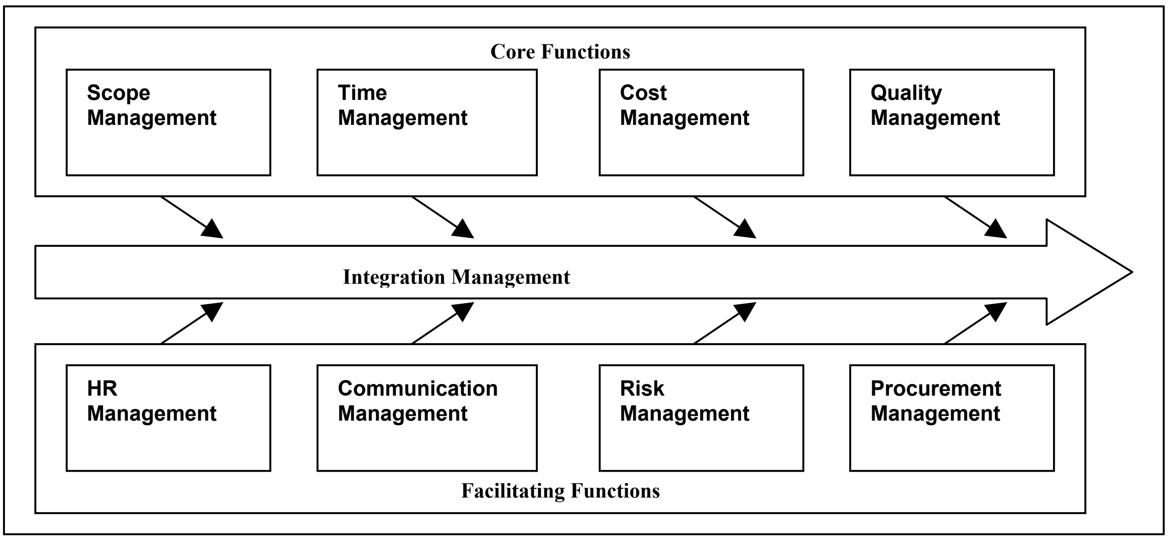


Figure 2.3: PMBOK® knowledge areas (Adapted from Schwalbe, 2004)

Each knowledge area is further divided into a number of processes. The total number of processes is 44 (PMBOK® Guide, 2006) and these are listed in appendix A.

2.4.1.2 Euromethod

Euromethod (EM) has been in use since November 1989 as version 0 within the European Union (EU) (Euromethod, 2002a). Its development has been the result of effort for the mutual understanding between customers and suppliers of information systems. EM version 1 was launched in July 1996. In order to promote EM outside the EU, it is now marketed under "Information Services Procurement Library" – ISPL (Euromethod, 2002b).

According to Hughes and Cotterell (2006), an organisation can be perceived as having a number of information systems and each one may need to be changed from time to time from one state (initial state) to another (final state). Each change is implemented by an adaptation, which transforms an information system from a defined initial state including knowledge into a defined final stage. This is shown in figure 2.4.

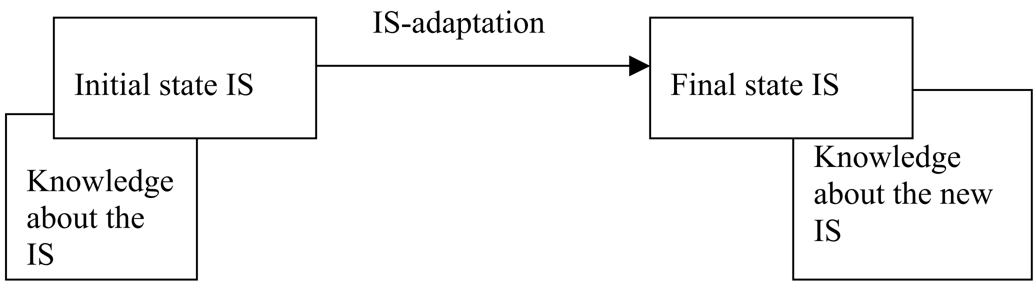


Figure 2.4: Information System adaptation

EM can be used in an information system involving:

(i) a customer,
(ii) a supplier (external or internal to the organisation), and
(iii) a contract (formal or informal) between the customer and the supplier.

2.4.1.3 BS 6079: 1996

BS 6079 ‘Guide to project management’ was published as a 49-page document in 1996 (Hughes and Cotterell, 2006). It is a set of guidelines rather than a standard. It has a traditional engineering rather than an IT background.

The main elements of BS 6079 are the project cycle, project organisation, planning process and project control and these are discussed below:

(i) A project cycle (depicted in figure 2.5).

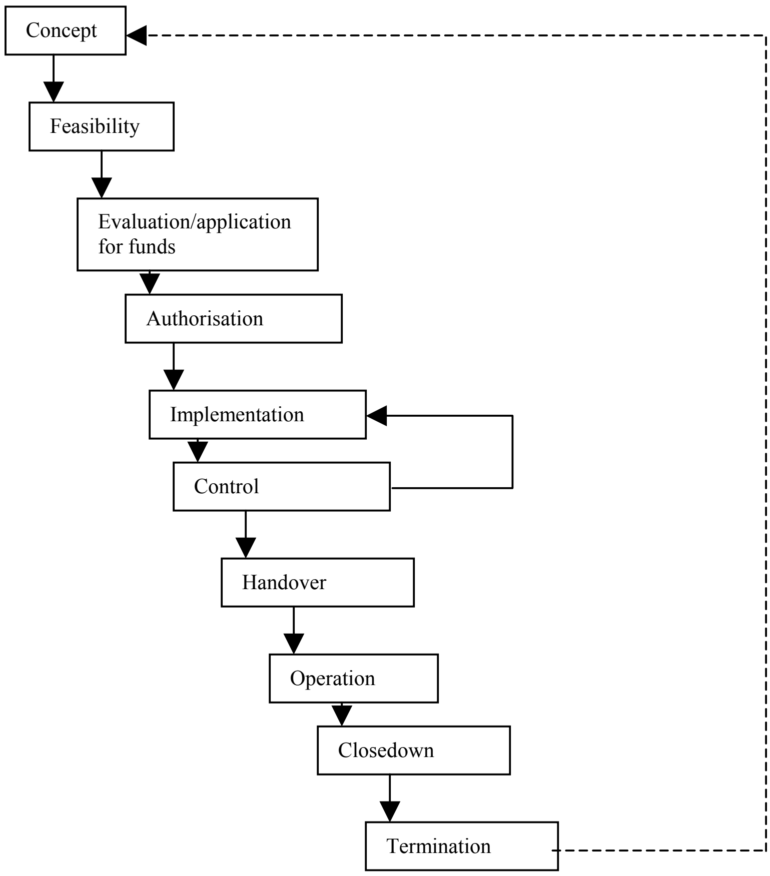


Figure 2.5: Project cycle (Hughes and Cotterell, 2006)

BS 6079 views a project as a system life cycle. The actual development of the project starts after the feasibility study has established that it is worthwhile and ends when the system is operational. A new cycle starts again after the system has been in operation for some time and no longer fulfils the objectives. The implementation and control elements follow an iterative loop so as to correct any deviation.

(ii) Project organisation.

A project is considered as a set of activities that cuts across the normal functional structures of most organisations. A matrix management structure is desirable. Explanation is provided on how to bring about an organisational change although this may take 3 to 5 years.

BS 6079 puts an emphasis on task owners, whereby each task has an owner, who is held accountable for its successful completion.

(iii) Planning process.

According to BS 6079, a plan should contain five main elements, namely:

- Introduction.
- Commitment acceptance.
- Work breakdown structure (WBS).
- Schedule.
- Statement of work (SOW).

The activity-based approach of BS 6079 places an emphasis on a WBS right from the beginning of the project. In addition, a statement of work is created for each element in the WBS.

(iv) Project control.

This exercise is carried out through:

- Change control.

  Although no detail steps are provided, an outline of the requirements is followed.

- Financial control.

  Disbursement of funds may be made as and when progress is achieved. However, funding must be allowed for contingencies.

- Monitoring progress.

  Reports from task owners are analysed at this stage. Risk assessment should be continuously made and use of techniques based on statistical probability is recommended.

- Supporting Techniques

  Calculation of the net present value (NPV) and the analysis of the earned value are encouraged at this stage.

2.4.1.4 ISO 12207

ISO 12207 has been developed specifically for software development projects (Hughes and Cotterell, 2006). It focuses on the characteristics of good documentation as lack of documentation is very often an issue complained of by software developers. Conversely its availability may result in reluctance to be read.

ISO12207 perceives a software development project as an Information System (IS). A project comprises a set of activities. Each activity passes information to the next activity. The effectiveness of the IS determines the success or failure of the project.

Five processes are identified:

(i) Acquisition.

(ii) Supply.
(iii) Development.
(iv) Operation.
(v) Maintenance.

The above five processes interrelate as in figure 2.6.

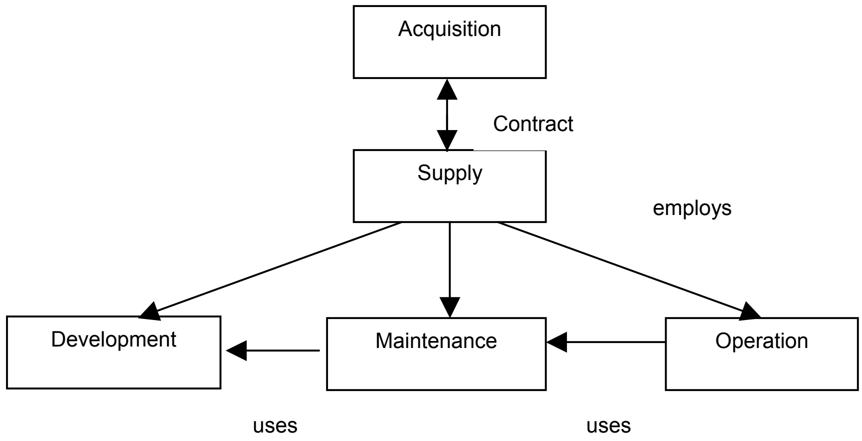


Figure 2.6: ISO 12207 processes (Hughes and Cotterell, 2006)

The maintenance and operation processes, as per figure 2.6, belong to the post-implementation phase. The Acquisition Process involves a set of procedures that the customer has to go through to obtain the requested software from a supplier. The Supply Process involves the supplier's decision to bid after receiving the Request For Proposal (RFP), preparation of a response, contract negotiation in case of a successful proposal, planning of the requirements based on the RFP, execution of the work, necessary control and finally the delivery of the product.

The development process involves the analysis of the requirements of the user, the overall design of the interacting components, coding, testing and installation/acceptance support.

2.4.1.5 PRINCE 2®

PRINCE 2® (PRojects IN Controlled Environments) is a structured project management methodology for managing projects in a logical, organised way, following defined steps (Hughes and Cotterell, 2006). PRINCE 2® was launched in 1996 and is now the de-facto standard for project management being used in both private and public sectors in the United Kingdom for all types and size of projects (Wideman, 2002; Key Skills Ltd, 2003).

The methodology as per PRINCE 2® Documentation/manuals, covers the following processes:

(i) Planning.
(ii) Directing a Project.
(iii) Starting up a Project.
(iv) Initiating a Project.
(v) Managing Stage Boundaries.
(vi) Controlling a Stage.
(vii) Managing Product Delivery.
(viii) Closing a Project.

The processes are interrelated as in figure 2.7.

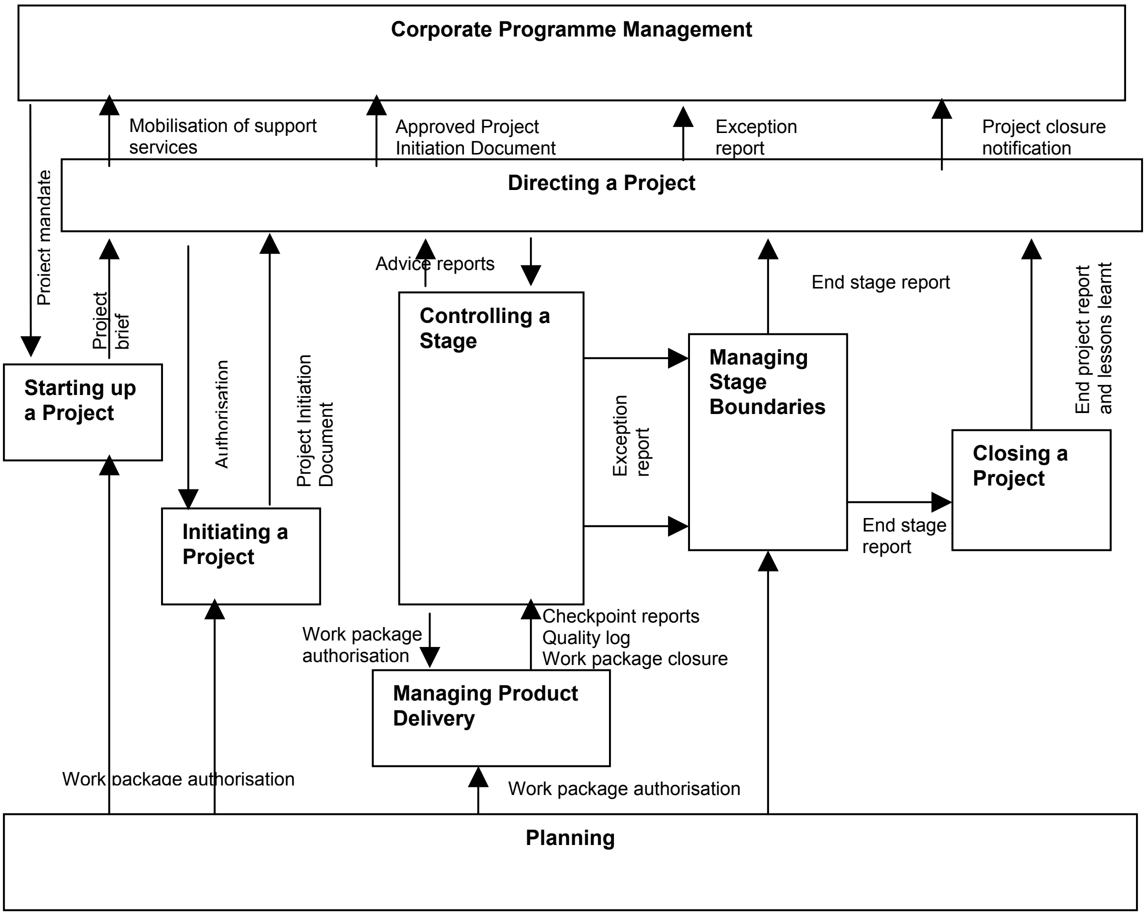


Figure 2.7: PRINCE 2® processes (CCTA, 1997)

The eight processes describe the steps from inception of a project to its closing stage. The processes do not represent a sequential step-by-step guide to project management and many of the steps are iterative while some can be done in parallel with others (CCTA, 1997). The processes are

- Planning.

  Planning is a repeatable process and used by the other processes whenever a plan is required.

- Directing a Project.

  Directing a Project runs from the start-up of the project until its closure. This process is aimed at the Project Board. The Project Board manages by exception, monitors via reports and controls through a number of decision points.

- Starting up a Project.

  This is the first process in PRINCE 2®. It is a pre-project process, designed to ensure that the pre-requisites for initiating the project are in place. The process expects the existence of a project mandate, which defines in high-level terms the reason for the project and what outcome is sought.

- Initiating a Project.

  The objectives of Initiating a Project are to:

  - Agree whether or not there is sufficient justification to proceed with the project.
  - Establish a stable management basis on which to proceed.
  - Document and confirm that an acceptable Business Case exists for the project.
  - Ensure a firm and accepted foundation to the project prior to commencement of the work.
  - Agree to the commitment of resources for the first stage of the project.
  - Enable and encourage the Project Board to take ownership of the project.
  - Provide the baseline for the decision-making processes required during the project's life.
  - Ensure that the investment of time and effort required by the project is made wisely, taking account of the risks to the project.

- Managing Stage Boundaries.

  This process provides the Project Board with key decision points on whether to continue with the project or not.

- Controlling a Stage.

  This process describes the monitoring and control activities of the Project Manager involved in ensuring that a stage stays on course and reacts to unexpected events. The process forms the core of the Project Manager's effort on the project, being the process which handles day-to-day management of the project.

- Managing Product Delivery.

  The objective of this process is to ensure that planned products are created and delivered by:

  - Making certain that work on products allocated to the team is effectively authorised and agreed.
  - Ensuring that work conforms to the requirements of interfaces identified in the Work Package.
  - Ensuring that the work is done.
  - Assessing work progress and forecasts regularly.
  - Ensuring that completed products meet quality criteria.
  - Obtaining approval for the completed products.

- Closing a Project.

  The purpose of this process is to execute a controlled close to the project. The process covers the Project Manager's work to wrap up the project either at its end or at premature closure. Most of the work is to prepare input to the Project Board to obtain its confirmation that the project may close.

2.4.2 Comparision of methodologies/models/standards

A comparison of the methodologies/models/standards presented above is as per table 2.1.

Table 2.1: Comparison of methodologies/models/standards

	PMBOK®	Euromethod	BS6079:1996	ISO12207	PRINCE 2®
Focus area	Reference document describing the generally accepted project management practices.	Primary concern is at the contractual level.	Set of guidelines on project management with the main focus a description of project management techniques.	Prime area of concern the documentation created and used by a software development project, especially when a supplier is involved.	Mainly associated with delivery of products.
Strengths	• Simple process model for project Management. • It covers procurement. • It covers the actual procurement, pre-assignment or negotiation for team members for a project in some detail. • It identifies needs to be covered in human resource management, and soft skills in general.	• Practical advice on how decisions are to be taken rather than stating what the decisions are.	• Written primarily for small to medium sized organisations. • Financial control is dealt with fully.	• Relates specifically to software development.	• Widely used de-facto standard for project management in the UK. as well as USA, Australia, New Zealand, The Netherlands, France, Italy, Hong Kong, South Africa, Croatia and Poland. • It offers a complete change control approach.

	PMBOK®	Euromethod	BS6079:1996	ISO12207	PRINCE 2®
Weaknesses	• Does not identify personal competencies and has no technique level detail. • There is little information about configuration management, and certainly offers no link between it, the Configuration Librarian role and change control. • It only mentions about the need for a complete change control approach.	• Sheer size of documentation. • The name (Euromethod) discourages organisations outside European Community to use it, fearing that it is meant for European countries. Its name was therefore changed to Information Services Procurement Library (ISPL).	• It is a guidance document rather than a statement of requirements for project management. • It is not intended for formal conformance but it can be used for contractual arrangements. • Has an engineering rather than IT background.	• Does not specify the details of how to perform the activities and tasks comprising the processes. • Not applicable to the purchase of commercial off-the-shelf software products.	• Does not include every aspect of project management such as team selection, motivation, which are often critical to the success of projects. BS6079 and PRINCE 2® can be used together within an organisation to provide a more complete coverage of the various aspects of project management. • It is generic and therefore does not specifically address problems in IT. • It is not – and was not designed to be – a comprehensive quality system. It does not fully address the requirements of ISO 9001 according to the CCTA PRINCE 2® manual. • It does not cover Project Human resources management. • Most of Project Procurement Management is not covered. PRINCE 2® regards this as a specialist activity, rather than a generic part of project management. • Cost control not dealt with in detail. • Sheer size of documentation is intimidating.

In addition to the comparative study in table 2.1, a further comparison exercise based on the principles of software project management was carried out and the results are presented in table 2.2.

Table 2.2: Comparison of methodologies/models/standards based on the principles of software project management

Principles \ Methodologies	PRINCE 2®	PMBOK®	BS6079	Euromethod	ISO 12207
Scope	Yes	Yes	Yes	Yes	Yes
Planning	Yes	Yes	Yes	Yes	Yes
Evaluation	Yes	Yes	Yes	Yes	Yes
Project development approaches	Yes	Yes	Yes	Yes	Yes
Software effort and cost estimation	Partially	Yes	Yes	Yes	Yes
Activity planning	No	Yes	Yes	Yes	Partially
Monitoring and control	Yes	Yes	Yes	Yes	Yes
Risk management	Yes	Yes	Yes	Yes	Yes
Resource allocation and control	No	Yes	Yes	Yes	Yes
Contract management	No	Yes	Yes	Yes	Yes
Human resource management	No	Yes	No	No	No
Quality management	Yes	Yes	Yes	Yes	Yes
Environmental (cultural, social, economic, political aspects) management	No	No	No	No	No

2.4.3 Project management maturity models

Some project management models having a set of maturity levels are used for the management of projects. Most of them have typically 5 maturity levels, each one comprising of some process areas (or focus areas), which are necessary for the management of projects. Project management maturity is a level of maturity to which an organisation has applied project management techniques and is using them in a proper and mature way (Supić, 2005). A project management maturity model provides the progressive development of an enterprise-wide project management approach, methodology, strategy and decision-making process (PM Solutions, 2004).

Several maturity models are available like MicroFrame self-assessment tools, Kerzner's maturity model, SEI's Capability Maturity Model Integration (CMMI), PRINCE 2 Maturity Model and Organizational Project Management Maturity Model (OPM3™). Most of them are defined by five maturity levels, namely 1 (Ad-hoc) through 5 (Optimising). The lowest level is typically in which organisations begin with. No project management processes and practices are consistently applied at this level.

An organisation has achieved full project management maturity (the highest level) when it has met the requirements and standards for project management effectiveness and is capable of demonstrating improvements such as on-time project delivery, cost reductions, organisational efficiency and profitability (PM Solutions, 2004).

The details of maturity models are described in chapter 5.

2.5 Conclusion

In this chapter, a definition of project management and a historical background of project management were provided along with the associated tools and techniques such as Gantt charts, Work Breakdown Structure, cost estimation and risk management were mentioned. Major projects that have marked the history of ancient civilizations were discussed. Modern project management was considered with projects like the Hoover Dam, the Apollo project, the development of the atomic bomb and the Internet.

With the revolution in IT as from 1980, project management acquired greater recognition. Project management software for personal computers was readily available and widely used. The period during which software project management emerged was found to be during the early 1980s.

The project management principles were discussed in terms of phases involved in managing software projects. Project management methodologies/models/standards were described with examples of the popular methodologies/models/standards in use. The different methodologies/models/standards were compared so that their shortcomings could be highlighted. The strengths and weaknesses of these methodologies/models/standards as well as to what extent these adhere to the principles of project management were presented. The available project management methodologies/models/standards were found to lack the feature to deal with the cultural, social, economic and political aspects in connection with software projects. These aspects are considered important for the success of projects in developing countries. These information are important to set the scene for further research in the quest for the choice of a project management methodology/model/standard for developing countries.

Maturity models were briefly described and the details are deferred to chapter 5. A study of these models was undertaken so as not to leave any stone unturned in the choice of a suitable project management methodology/model/standard as maturity levels are related to the level of success of organisations.

In the next chapter, project management studies carried out in developed as well as developing countries are presented so as to give more prominence to the need for a suitable project management methodology/model/standard for the latter countries. The reader is exposed to the SPM problems facing these countries that will set the scene for further research.

Chapter 3

Status of IT/Software Project Management in the US, UK and African countries

---

Table of contents

3.0 Chapter layout

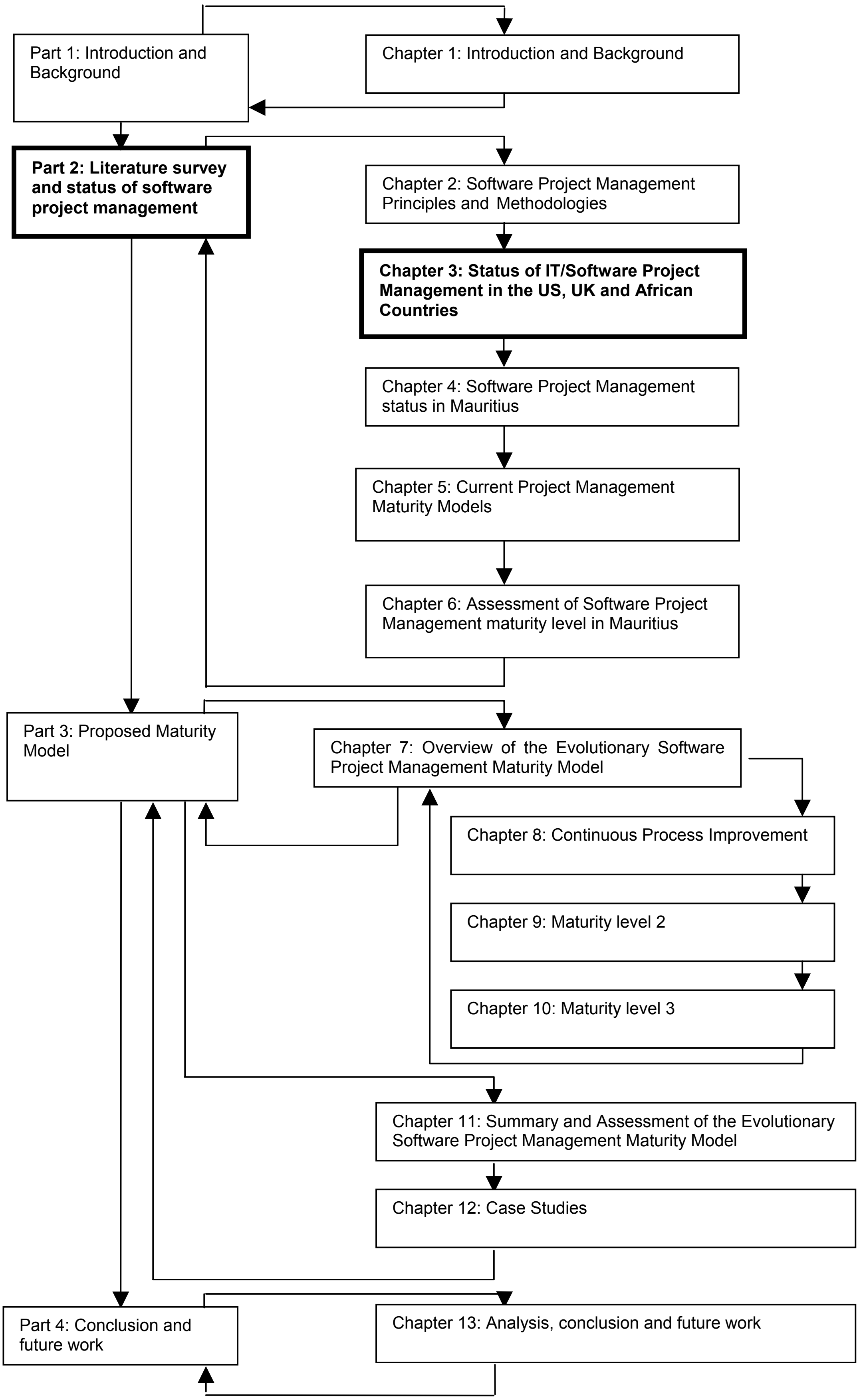


Figure 3.1: Chapter 3 in context within the overall thesis

3.1 Introduction

Although many software project management tools and techniques are applicable to both general project management as well as IT and software project management, the latter most particularly pertaining to software development requires special attention. It must be noted that IT and software project management borrowed much of its concepts from traditional engineering project management (dealing with construction and engineering areas).

In this chapter, an analysis ofthe state of IT and software project management in the UK and the US is provided. The success rates of IT projects in these countries are highlighted. In addition, the situation of project management with special focus on IT and software project management is also discussed regarding African developing countries. The differences in social and economic conditions of these countries as compared to developed nations are highlighted. The applicability of IT and software project management approaches of Western origin for African nations is discussed.

3.1.1 Aim and structure of this chapter

In this chapter, an analysis is carried out regarding the status of project management, with special focus on IT and software project management in the US, UK and African developing countries. The CHAOS report (Standish Group, 2004) is considered for the US, while the Computer Weekly (Huber, 2003) is chosen to explore the status of IT and software project management in the UK.

In addition, several studies are selected to show the status of project management in African developing countries. The specificities of IT and software project management are highlighted during the discussions.

The remainder of the chapter is structured as follows:

Section 3.2 - In this section, the state of IT and software projects in the US and UK is discussed. Problems related to IT and software projects in developed countries like the US and the UK are also highlighted.

Section 3.3 – The state of IT and software project management in African developing countries is discussed in subsections 3.3.1 to 3.3.5 as elaborated below:

Section 3.3.1 – In this section, the studies carried out by Rondinelli (1976) are analysed in terms of problems related to IT and software project management although the findings of the author are applicable to all types of projects. Solutions are also proposed to overcome the obstacles identified.

Section 3.3.2 – This section comprises the studies carried out by Stuckenbruck and Zomorrodian (1987), who highlighted the problems related to project management in developing countries and proposed solutions. Information relevant to software project management is extracted for the purposes of this thesis.

Section 3.3.3 – In this section, the findings and a methodology put forward by Ronan (1993) for managerial effectiveness are discussed. The methodology can help organisations to assess their status and determine what improvement is required. Although the proposed methodology is devised for management in general, the application to IT and software project management in particular may prove to be useful.

Section 3.3.4 - In this section, the studies carried out by Muriithi and Crawford (2003) are summarised and analysed critically. The results form an important basis for further work in developing countries aspiring to develop the software industry, both as a supporting sector or a pillar of the economy. Muriithi and Crawford (2003) provided important information on the situation of African developing countries.

Section 3.3.5 – Sonnekus and Labuschagne (2004) focused on the state of IT and software project management in South Africa and the results of their findings are discussed in this section. Their findings show the specificity of problems encountered in South Africa. In spite of the problems highlighted by the authors, South Africa has been able to emerge as a successful country as regards to IT and software project management by achieving a high percentage of successful projects.

Section 3.4 – As a conclusion, the specificity of problems related to IT and software project management in African developing countries are highlighted. However, the aim of this thesis is not to investigate specific problems of each African country. The case of Mauritius is considered for further research in this thesis. The status of IT and software project management in Mauritius is, therefore, covered in chapter 4. The analysis of the results may, therefore, determine any need for enhancement in IT and software project management principles, tools and techniques for Mauritius or even show the way forward to improve the success rate of software delivery.

3.2 The state of IT and Software Project Management in the US and the UK

With the proliferation of software projects, worldwide efforts are being deployed to improve the management of IT and software projects. In this context, the Standish Group (2004) conducts research and publishes regularly an evaluation of IT and software projects. Table 3.1 shows the percentage of IT and software projects that succeeded since 1994 (Standish Group, 2004).

Table 3.1: Successful projects

Year	Successful projects (%)
2004	29
2002	34
2000	28
1998	26
1996	27
1994	16

It is observed that since 1996, for every three IT projects undertaken, on average only one succeeds (28.8%). The CHAOS (Standish Group, 2004) report identified ten success factors, although they claim that very few IT and software projects would require all ten factors. These factors are:

- *Executive support* - Lack of executive support hampers the process and progress of a project.
- *User involvement* - A project delivered on time and budget can fail if it does not satisfy user requirements. It is important to involve users as early as possible in a project.
- *Experienced project manager* - An experienced project manager contributes to 97% of successful projects.
- *Clear business objectives* - Clear business objectives eliminate ambiguities in a project. Every team member needs to be aware of the objectives to be achieved.
- *Minimised scope* - Minimised scope reduces the time taken to complete a project and, therefore, increases the chance of success.
- *Standard software infrastructure* - The use of standard software infrastructure avoids failure due to the integration of existing applications.
- *Firm basic requirements* - A minimal set of features can allow users and executive sponsors to see results quickly.
- *Formal methodology* - Consistency across projects is maximised with a formal methodology. According to the CHAOS research, 46% of successful projects used a formal project management methodology.

- *Reliable estimates* - Estimation in software development is a difficult task. Therefore, IT managers need to use their experience to provide realistic estimates.
- *Other criteria* - There are other factors that are important for the successful completion of projects. These include incremental milestones, proper planning and ownership.

A survey was also conducted based on data from 1,500 practicing IT and software project managers in the UK between October 2002 and January 2003 (Sauer and Cuthbertson, 2003). It was found that about 16% of the projects met their targets, which was no better than in the United States although the budget overrun, schedule overrun and underachievement on scope/functionality were respectively 18%, 23% and 7%.

The report claimed that the UK organisations had not yet achieved a superior and industry-wide project management capability. A comparison of abandoned, challenged and successfully completed projects is depicted in the table 3.2.

Table 3.2: Comparison on basic performance categories

	Abandoned	**Challenged**	**Successfully completed**
Standish Group 1995	31%	53%	16%
Standish Group 1999	28%	46%	26%
Standish Group 2003	15%	51%	34%
Computer Weekly 2003	9%	75%	16%

The results of the survey were published on the Computer Weekly website (Huber, 2003).

Project management is assisted by the use of some approaches, as briefly mentioned in chapter 2, in Western and European countries. These approaches are constantly improved to facilitate in the delivery of more successful projects.

As discussed above, the status of IT projects in the US and the UK revealed a low percentage of success (34% and 16% respectively in 2003 – see table 3.2).

From 1996 to 2004 (as shown in table 3.1), an average of 1 in every 4 (23%) projects was found to be successful. Although the US and UK have developed a number of tools and methodologies/models/standards for IT and software project management, still the percentage of successful IT and software projects remained low.

3.3 The state of IT and Software Project Management in African Developing Countries

African developing countries have major challenges to overcome as they are faced with different social and economic conditions as compared to the UK and the US. The subsections that follow highlight the problems encountered and the state of readiness to handle IT and software projects using methodologies/models/standards, tools and techniques that are prevalent in developed countries.

3.3.1 Analysis of studies carried out by Rondinelli

Rondinelli (1976) studied the problems of project management in developing countries leading to the failure of projects of a general nature. These reasons are still valid in the current context for African developing countries. IT and software projects are influenced by these problems and details are provided in the Mauritian context in chapter 4.

According to Rondinelli (1976), less developed countries are faced with serious problems in project execution pertaining to ineffective planning and management. In addition, project management practices as well as complex techniques and procedures in advanced countries have been prescribed for developing nations without taking into consideration their cultural, political and social traditions. As a result, a range of project management problems occurred and those most related to software project management are:

- **Ineffective project planning and preparation**.

  Governments are unable to allocate available resources to feasible projects as a result of inadequate capital planning and budgeting systems. Projects are executed with insufficient analysis, sectoral assessment, feasibility studies and technical appraisal. As the saying "failing to plan is planning to fail" is applicable here, failures are guaranteed for such projects. In case efforts are made to complete these projects, serious budget and time overrun may occur.

- **Faulty appraisal and selection processes**.

  This project management problem is related to unclear objectives and outputs of projects, emphasis on funding available rather than productive outputs while selecting projects, difficulty in cost estimation and omission of social, cultural and environmental impacts. Projects are also affected by long waiting periods in the processing and approval of projects.

- **Defective project design**.

  This type of problem refers to project designs that are inappropriate for local conditions. Project designs put forward for developed countries may not be applicable to developing or less developed countries like African countries.

- **Problems in start-up and activation**.

  The start-up of projects often suffers from delays in approval, inter-ministerial rivalries and lack of cooperation in allocating resources. Lack of technical assistance during project activation and failure to redesign the project when obstacles are discovered also result in project failures. Therefore, to increase the probability of success, a high level commitment is required from authorities concerned from the start of a project.

- **Inadequate project execution, operation and supervision**.

  This problem is related to insufficient capacity or competence of contractors and a lack of adequately trained and competent project managers. Inadequate salary structures in developing African countries also contribute to project failures, most probably through a lack of motivation of the project team. Rondinelli (1976) also attributed project failures to a lack of indigenous management skill. Instead, management skills from developed countries are being applied without prior studies to developing countries. Therefore, in order to address this project management problem, appropriate training needs to be provided to project managers, the salary structures of project managers in the software sector also need to be revised and indigenous management skills need to be encouraged.

- **Deficiencies in diffusion and evaluation of project results and follow-up action**.

  Inadequate monitoring and control of projects were noted by Rondinelli (1976). It was also found that training and retaining of personnel following a project completion exercise were difficult. The employees were probably using their experience to bargain for higher salaries in other organisations, thereby leading to high mobility of labour. In addition, failure to anticipate, plan for or adjust to the political and social impacts of projects on the local population was also the cause of failure of many projects in developing countries. Therefore, organisations concerned were required to identify projects that are citizen-centric.

Although the causes of success and failures of projects in developed countries must not be neglected, it is imperative for developing African countries to take cognizance of the problems of the local context. Therefore, the social, economic, political and cultural aspects as well as the other relevant factors identified by Rondinelli (1976) need to be taken into consideration. According to Rondinelli (1976), techniques, constraints and opportunities that are sensitive to national needs must be developed. The experiences of local officials need to be tapped as they are the people who know best what ought to work well in their countries.

3.3.2 Analysis of studies carried out by Stuckenbruck and Zomorrodian

Stuckenbruck and Zomorrodian (1987) explored the usefulness of project management approaches for developing countries. In their studies, project management for all types of projects are discussed. However, for this thesis, as far as possible only issues related to IT software project management are considered and the specific problems of African developing countries are extracted for the purposes of this thesis. Although the research of Stuckenbruck and Zomorrodian (1987) has been conducted a long time back, one can find that the problems faced by developing countries are still relevant as was confirmed by Muriithi and Crawford (2003). The findings of Muriithi and Crawford (2003) are discussed in section 3.3.4.

According to Stuckenbruck and Zomorrodian (1987), developing countries are faced with specific social, political and economic problems, although their severity differs greatly from country to country. Some of the problems identified could be summarised as:

- Widespread poverty.
- Unemployment.
- Low national income per capita.
- Shortage of investment capital.
- Widespread illiteracy.
- Heavy concentration of political power and lack of public participation in policy decisions. Certain political and social settings are also prone to bribery and corruption and this affects the spirit of developing the software sector and the inflow of foreign investment and assistance.
- Shortage of skilled manpower. The scarce resources in the software sector are strongly felt by developing countries as this sector is known to undergo dynamic changes over small periods of time.
- Administrative incapacity. The situation is exacerbated by the existence of rigid bureaucracies. There is also a lack of awareness about the philosophy of project management on the part of administrators. This results in the lack of effective project management methods in approaching problems.

Stuckenbruck and Zomorrodian (1987) put forward strong arguments for promoting project management in developing countries as follows:

- It attempts to achieve planned objectives within specific time and cost limits through the optimum use of available resources. An integrated planning and control system is also put in place while executing projects. Timely completion of projects within budget, as targeted by project management procedures, gives people more trust and confidence, especially in projects executed for the Government. Effective and efficient use of scarce resources is also favoured by employing project management tools and techniques.
- It enhances teamwork and encourages motivational leadership. Intensive interaction between team members is practised. Stuckenbruck and Zomorrodian (1987) pointed out that project management is a way of getting people to work effectively together.
- A range of tools (for example CPM and PERT) are available for the planning and tracking of project activities and tasks. Multiple projects can also be managed with greater dexterity. Software tools are also being continuously enhanced and software professionals are those who benefit the most.
- Given that project management is result-oriented, the possibility of projects becoming part of a bureaucratic system is avoided to some extent.

Stuckenbruck and Zomorrodian (1987) asserted that the issue of culture is one of prime focus while transferring any management technique or model from one country to another and especially when developing countries are concerned. They also referred to the cultural aspects using Hofstede's dimensions (Hofstede, 1983; Muriithi and Crawford, 2003), namely individualism versus collectivism, uncertainty avoidance, power distance and masculinity versus femininity. The example of Japan as a successful country with the incorporation of cultural factors into its management system was highlighted (Ohashi High Technology Corp., 2007). Furthermore, in Japan, emphasis on groups of people could be observed together with interdependence, subordination and obligation towards others.

According to Korton (1980), as cited by Stuckenbruck and Zomorrodian (1987), many Western management practices were developed to improve the already well-developed organisational settings. These practices were then transferred to developing countries having less well institutionalised environments, thereby causing disturbances in the management of projects. According to the Korton (1980), the strategy for implementing project management in developing countries must be consistent with the culture and the characteristics of the particular society and the configuration of its economic, political and administrative system. The introduction of any project management framework into an organisation requires the following to be carried out as per the recommendation of Stuckenbruck and Zomorrodian (1987):

- Plan for the implementation of project management in order to minimise the organisation's disturbance.
- Select the project manager carefully.
- Identify the local needs and conditions.
- Evaluate the benefits and costs of using the selected project management approach.
- A first pilot project has to be chosen carefully instead of applying the project management approach to all projects. The project should be a safe and simple one instead of choosing a complex one where the chance of successfully applying the project management approach is low.
- The project management approach needs to be sold to all stakeholders so as to gain the required support from all of them. At the same time, threats related to job security, income, careers and reputation should be reduced or eliminated. An awareness campaign needs to be conducted to inform all those concerned that the project is a vehicle for change. The environment of successfully implementing any proposed project management framework can constitute a breeding ground for the staff in

terms of experience that would be useful for future career opportunities. Consequently, an effective means of bringing about a reform would be achieved.

According to the studies carried out by Stuckenbruck and Zomorrodian (1987), the use of project management principles is undoubtedly a solution for developing countries. However, the transfer of Western practices and tools may cause the failure of projects due to political, organisational and cultural obstacles. Every nation has a different cultural heritage and customs, which require different approaches. Modification or customisation of an approach is required so that it fits the environment in which it is applied. Therefore, a project management framework that incorporates some indigenous components has a better chance of success.

3.3.3 Analysis of studies carried out by Ronan

Ronan (1993) suggested the need for a framework or model to facilitate organisational effectiveness in Africa. He pointed out the inappropriate educational systems on the continent and the transfer of training systems from the Western world without taking cognisance of the cultural settings of the host country. The proposed managerial effectiveness matrix (figure 3.2) as an operational model of management development for African managers indicates the skills required for project managers and administrators.

According to Ronan (1993), African managers have a high need for relationships and status and a low need for the achievement of objectives. He also pointed out that Western models have failed to deliver desired results. The model proposed by Ronan (1993) seeks to combine the best features of Western management thinking while accommodating the African need for relationships. In this context, a combination of hard skills (comprising the technical and strategic skills) and soft skills (comprising the interpersonal and political skills) has been incorporated in his model. According to Ronan (1993), the exclusive use of hard skills would tend to have an internal orientation and focus on efficiency, while the exclusive use of soft skills would tend to have an external orientation and focus on effectiveness. The managerial effectiveness matrix is shown in the figure 3.2 (adapted from Ronan (1993)).

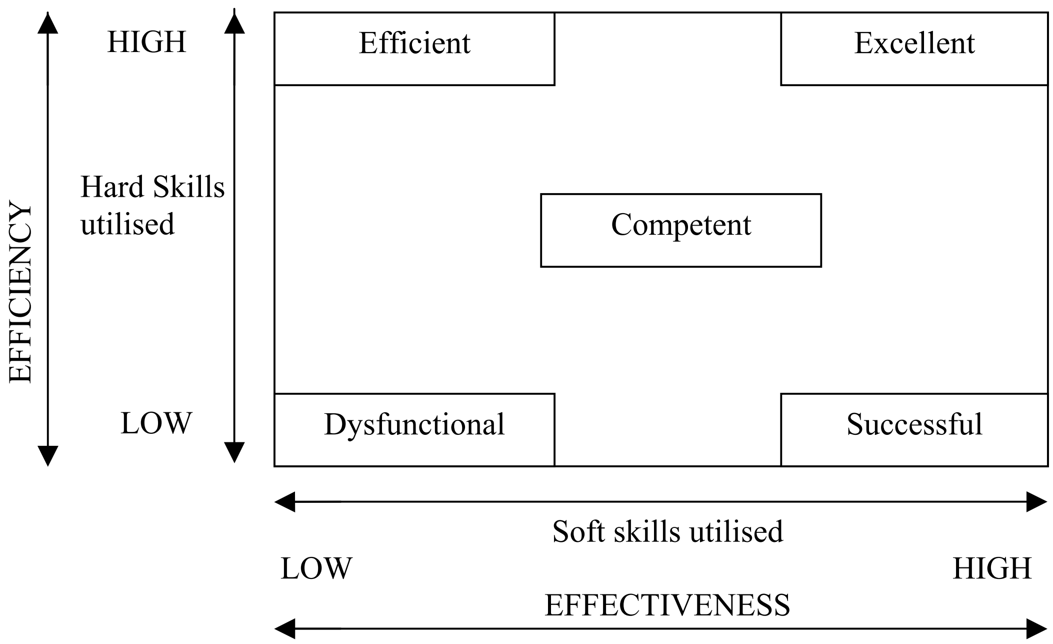


Figure 3.2 Managerial effectiveness matrix (Ronan, 1993)

According to Ronan (1993), the five managerial types available in the managerial effectiveness matrix are:

- The Dysfunctional Manager. This type of manager is depicted as the poor performer and is characterised by low use of both hard and soft skills. African managers possess such a style due to a lack of motivation and low remuneration. Such managers may have been promoted due to political patronage in spite of their low capabilities.
- The Competent Manager. This type of manager is the average performer and it is hypothesised that the majority of African managers fall into this category.
- The Excellent Manager. This type of manager is characterised by the high use of both hard and soft skills. The excellent manager is a business leader with strategic vision and is a great motivator. According

to Ronan (1993), few African managers fall into this category. However, competent managers could be promoted to this category through appropriate training programmes.

- The Efficient Manager. This type of manager scores high on hard skills utilisation and low on soft skills utilisation. Therefore, the efficient manager is unable to practice the interpersonal and political skills required in the successful performance of his/her duties.
- The Successful Manager. This type of manager scores high on the use of interpersonal and political skills and is perceived by his/her superiors as performing at an excellent level. Ronan (1993) hypothesised that this type of manager is the principal contributor to the ineffectiveness of many African organisations. The successful manager relies on political skills for survival and often calls on highly skilled subordinate managers for assistance in order to stay in office, otherwise his/her performance is rated as dysfunctional.

The managerial effectiveness matrix proposed by Ronan (1993) allows managers to be classified into five categories for eventual management development. The matrix may help developing African countries in the management development of project managers in the software sector by emphasising on the required skills for a successful delivery of projects. As Ronan (1993) also pointed out, the cultural aspects of the country must not be ignored by such managers in the performance of their duties. Any project management framework must be carefully selected and customised (although an indigenous approach would be preferable) while at the same time using a good blend of hard and soft skills to manage projects.

3.3.4 Analysis of studies carried out by Muriithi and Crawford

Muriithi and Crawford (2003) investigated the applicability and relevance of project management approaches, tools and techniques in African countries, including the islands of Madagascar, Mauritius and Seychelles. Although they concentrated on general projects, the applicability of software projects is considered in this section while we address SPM in chapter 4. Project management approaches of Western origin were found to be unsuitable for African countries. These countries have different cultural, social, economic and political conditions, all of which have an impact on the management of projects. These conditions are discussed briefly in this section.

Cultural aspects were represented by Hofstede's dimensions (Hofstede, 1983; Muriithi and Crawford, 2003), namely power distance, uncertainty avoidance, masculinity/femininity and individualism/collectivism. The definition of each dimension is given below:

- Power distance. This is the degree of acceptance of instructions or decisions from more powerful individuals by less powerful individuals in society. A high power distance, therefore, does not favour open criticism and results in tall organisational structures.
- Uncertainty avoidance. It is the extent to which ambiguity is perceived as threatening and risk-taking behaviour is avoided.
- Masculinity/femininity is the extent to which masculine traits such as achievement, courage and competition are valued over feminine values and behaviour such as nurture and sympathy.
- Individualism/collectivism is the extent to which people define themselves as individual entities or in terms of groups.

Based on the above Hofstede's dimensions, table 3.3 (Muriithi and Crawford, 2003) shows the characteristics of African countries in general as compared to the United Kingdom.

Table 3.3: Hofstede's dimension

Hofstede's Dimensions	African countries	UK
Power distance	High	Low
Uncertainty avoidance	Moderate	Low
Masculinity/femininity	Moderate masculinity	Masculine
Individualism/Collectivism	Collectivist	Individualist

Social conditions in African countries are characterised by:

- Extended family ties. Individuals are not only committed to organisations but also to their families.
- Young and rapidly expanding populations.
- Overcrowded urban centres.

The economic conditions in African countries are characterised by:

- Declining real wages, security and stability of employment.
- Low per capita incomes.
- Rising unemployment levels.
- Struggle of employees to supplement their earnings through informal part time employment.
- A trend to become poorer.

In general, African countries were found to have weak and unstable political institutions. Political power resides in a smaller elite. According to Muriithi and Crawford (2003), managers behave as though they are running kingdoms. Projects deemed to provide political/constituency benefits are given priority. Even managerial success requires political skills.

The management of a project through a life cycle approach, from an African perspective, covers the following:

- *Initiating and designing projects.*
  The high power distance that exists within organisations requires novel ways of facilitating participation of employees lower down the hierarchy.
- *Implementation.*
  The scarcity of project resources forces top management to be more concerned with the allocation of such resources rather than monitoring of the budget. A proper communication framework upwards in the hierarchy can also lessen uncertainty avoidance during this phase of project development. Human factors are also not to be neglected, particularly with respect to project team motivation and contract administration.
- *Closing/termination.*
  The boundary between the operational phase and the sustainability of this phase tends to be hazy. This problem is the consequence of a lack of employment opportunities and these lead to projects that drag on, thereby causing an operational budget overrun.

The management of major project management functions, from an African perspective, is characterised by (Muriithi and Crawford, 2003):

- *Project integration and scope management.*
  A slow delegation process of top management makes the project integration function difficult. Given the political interest in projects with public/community benefits, the ability of project managers to liaise directly with politicians are often solicited.
- *Project organisation and communication.*
  High power distance may be disruptive to organisations should there be no proper and efficient communication infrastructure in place. Informal networks have to be exploited to overcome the prevailing high power

distance and the use of family/community networks needs to be considered for projects' benefit.

- *Risk avoidance.*
  Risks are normally avoided due to a high uncertainty avoidance. Therefore, organisations do not tend to innovate. They prefer to follow what others have done successfully in a similar sector.
- *Procurement and quality.*
  The use of local appropriate technologies is welcomed to ensure contract compliance and the required level of quality. There is a need to focus on local values instead of showing a preference for overseas suppliers/contractors.

Hofstede's four dimensions namely power distance, uncertainty avoidance, masculinity/femininity and individualism/collectivism coupled with weak and unstable political institutions as well as difficult economic situations characterised by poverty, show that not everyone is ready for the introduction of formalised and structured management techniques into their work habits (Muriithi and Crawford, 2003). They concluded that there was an urgent need for research work in this area to formalise a new framework and confirm the necessary tools and techniques for the management of general projects in African countries.

3.3.5 Studies carried out by Sonnekus and Labuschagne

Inspired by research conducted in the US (in essence by the Standish Group), Sonnekus and Labuschagne (2004) carried out a survey to establish the relationship between IT project management maturity and IT project success in a South African context. Schiltz (2003) also showed the relationship between project management maturity and project success.

Sonnekus and Labuschagne (2004) used a combination of interviews and questionnaires to gather reasons for success in IT and software projects as well as to assess the level of maturity reached by South African IT industries. IT and software or general project management models are classified into sequential and maturity models in this thesis (see chapter 2). A noteworthy feature of the research of Sonnekus and Labuschagne (2004) lies in the mapping of sequential models onto maturity models. In this way, the maturity levels of all selected South African industries have been assessed and the relationship to IT and software project success has been objectively determined. The reasons for project success were found to be different from that of the US as per the table 3.4 reproduced from the paper of Sonnekus and Labuschagne (2004).

Table 3.4: Reasons for project success

Ranking	**Reasons for project success according to interviews carried out in South Africa**	**Reasons for project success according to CHAOS 2000 report carried out in the US**	**Ranking**
1	Auditing of processes	Minimised scope	6
2	Business objectives	Clear business objectives	1
3	Change control processes	Reliable estimates	7
4	Communication infrastructure	Standard infrastructure	9
5	Executive support	Executive support	2
6	Formal methodologies	Formal methodology	5
7	Handling change		
8	Project manager competency/experience	Experienced project manager	3
9	Project team	Skilled staff	8
10	Requirement definition	Firm basic requirements	4
11	Support of innovative / new technology	-	-
12	Understanding user needs	-	-
13	User involvement	User involvement	10
14	User understanding of technology	-	-

The above results, therefore, show that reasons for project management success may not be universally consistent across countries.

IT and software project management maturity was determined by Sonnekus and Labuschagne (2004) by taking into consideration the nine knowledge areas of the PMBOK® (Scope management, Time management, Cost management, Quality management, Human resource management, Communication management, Risk management, Procurement management and Integration management). The knowledge areas are grouped into core functions and facilitating functions, all of them tied together by the integration management knowledge area as shown in section 2.4.1.1 of chapter 2. Sonnekus and Labuschagne (2004) defined maturity levels 1 to 5 as follows:

Level 1 **– Initial level**, whereby processes are disorganised or chaotic.

Level 2 **– Repeatable level**, whereby existing processes are not considered an organisational standard.

Level 3 **– Defined level**, where organisational standards and institutionalised process are defined.

Level 4 **– Managed level**, project management processes, standards and supporting systems are integrated with other corporate processes and systems.

Level 5 **– Optimising level**, where the organisation uses feedback from established processes to continually improve and redefine them.

The maturity levels (1-5) for the different knowledge areas as determined by the research are shown in table 3.5.

Table 3.5 Maturity levels based on PMBOK® Knowledge Areas

PMBOK® Knowledge Area	**Maturity level of projects in SA according to Sonnekus and Labuschagne (2004)**
Integration Management	3.02
Scope Management	3.12
Time Management	2.99
Cost Management	2.94
Quality Management	2.82
Human Resource Management	2.87
Communication Management	2.93
Risk Management	2.65
Procurement Management	2.99

From table 3.5, the average maturity level of South African IT industries was found to be 2.92 (calculated from the average value of the nine knowledge areas in table 3.5) on the scale 1 to 5. As compared to US IT industries, South African IT industries are better positioned as far as the percentage of successful projects are concerned, but there is much room for improvement so as to attain the ambitious level 5.

The knowledge areas are further subdivided into 39 processes (see appendix A) and each process was sub-divided into five process groups, namely:

- Initiating process group.
- Planning process group.
- Executing process group.
- Controlling process group.
- Closing process group.

Furthermore, Sonnekus and Labuschagne (2004) also determined the maturity per process group. The average figures are shown in table 3.6.

Table 3.6 Maturity levels based on Process Groups

	Process Groups				
	Initiating	**Planning**	**Executing**	**Controlling**	**Closing**
Average maturity	3.27	2.94	3.00	2.92	2.93

According to Sonnekus and Labuschagne (2004), planning, controlling and closing process groups seem to require more attention on the part of South African IT industries. The improvement in the process groups could address project failures to some extent.

Sonnekus and Labuschagne (2004) came to the following conclusions:

- Improvement in informal communication was required in order to bridge the gap between technical and business staff. Better understanding of the needs of customers was found to be important to deliver better products and services.
- Risk management was not addressed adequately and this aspect contributed to project failures.
- Control processes are to be given greater consideration as they score the minimum value out of the five process groups.

An insight into the maturity of South African IT industries highlighted the areas that require attention to improve the percentage of successful projects. Sonnekus and Labuschagne (2004) also analysed the reasons for project success and failures in terms of the percentages of projects. Since a link between project success and project maturity was established by Sonnekus and Labuschagne (2004), it follows that every IT industry ought to concentrate on the relevant knowledge areas to increase the success rate of projects.

3.4 Conclusion

In this chapter, the status of IT Project management in the US and UK was discussed. It was found that in the US, on average one in every three projects was successful. The percentage of successful projects in the UK was even lower. This shows that even developed countries are facing difficulties to be successful in the field of IT.

Several studies related to the African developing countries were also considered. These countries have a high power distance, moderate uncertainty avoidance, moderate masculinity and a collectivist nature of the population. The study conducted by Muriithi and Crawford (2003) regarding Hofstede's dimensions was found to have an influence on general project management. This also implies that IT and software project management is by extension affected by Hofstede's dimensions, thus leading to even greater difficulties than developed countries to achieve an acceptable level of project delivery.

The status of IT and software project management in the South African context was discussed in terms of the maturity of the industry. It was noted that there was ample room for improvement as South Africa had not yet reached maturity level 3 on the scale 1 to 5. Given the issues raised by Muriithi and Crawford (2003), South Africa also needs to consider the cultural, social, economic and political aspects in addition to specific project management knowledge areas. This study paves the way to explore the status in other developing countries.

The next chapter brings Mauritius into perspective by highlighting its status of software project management.

Chapter 4

Software Project Management Status in Mauritius

Table of contents

4.0 Chapter layout

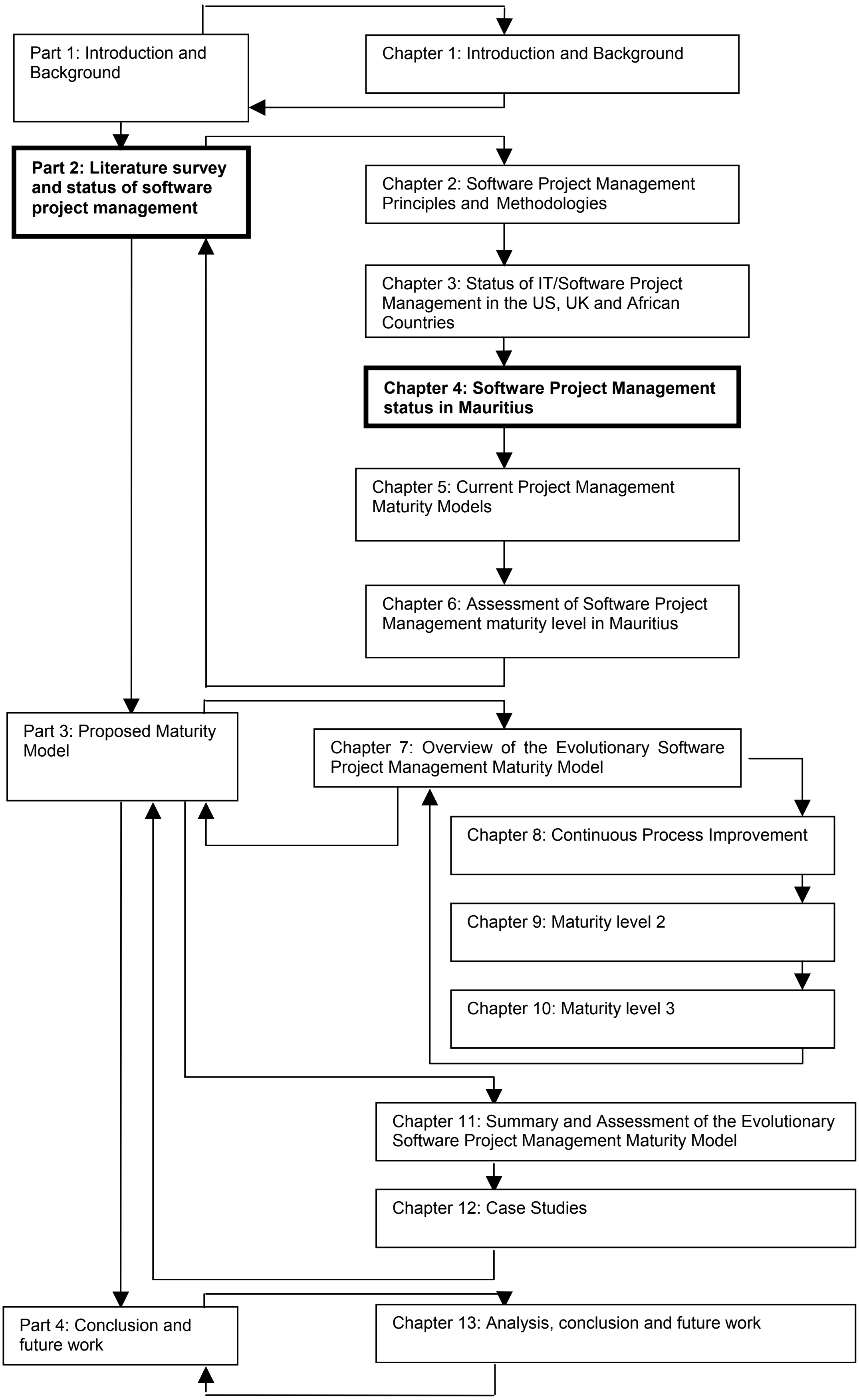


Figure 4.1: Chapter 4 in context within the overall thesis

4.1 Introduction

In the previous chapter, the status of project management in the US and UK were highlighted. The social, economic, cultural and political status of African developing countries were also discussed in the context of software project management. Mauritius is considered as part of these developing countries according to Muriithi and Crawford (2003).

Therefore, in this chapter, the results of a survey conducted in Mauritius (Sukhoo et al., 2004b) are also discussed. These results are vital in providing insights into the readiness of Mauritius to develop the software sector further in order to contribute towards the achievement of the Government's vision of transforming the island into a Cyber Island in the region.

4.1.1 Aim, objectives and structure of this chapter

The aim of chapter 4 is to analyse the results of surveys conducted in Mauritius regarding the status of software project management. Two surveys are discussed by Sukhoo et al. (2004b) and Boodhun (2005) respectively. Problems related to software project management in Mauritius are identified and solutions are also proposed. These studies form part of a larger research to determine any improvements required in the sector for the country to emerge successfully with IT as a pillar of the economy.

The deliverables, in terms of objectives, are the status of and suggested solution to improve software project management in Mauritius. These objectives are, however, intermediate ones in the quest to develop the IT sector further.

The remainder of the chapter is structured as follows:

Section 4.2 – In this section, the results of the survey conducted by Sukhoo et al. (2004b) on project management tools, techniques and methodologies used in Mauritius are presented and analysed.

Section 4.3 – This section comprises the studies conducted by Boodhun (2005) regarding the status of software project management in the public sector when he affirmed the need to apply project management techniques in order to meet target dates, improve the working relationship with the users in various Ministries as well as boost the morale of the staff.

Section 4.4 - Shortcomings pertaining to the requirements of software project management methodologies/models/standards are identified through the surveys conducted. Conclusions are drawn on the need to proceed further with research in Mauritius to determine what best suits the requirements of the country.

4.2 Survey of software project management tools, techniques and methodologies in Mauritius

The survey was conducted by Sukhoo et al. (2004b) and the results were published in the proceedings of the 2004 Project Management South Africa (PMSA) International Conference. The subsections below provide the details of the research.

4.2.1 Aim of the survey

The aim of the survey was to assess the status of software project management in Mauritius. As far as the author is aware, no such survey was conducted so far in this sector in the country. It was, therefore, thought that it would be a useful initiative to gather relevant information that could help in assessing any gap in software project management requirements. The data were subjected to statistical analysis.

4.2.2 Research Methodology

Two questionnaires were prepared, based on a literature search on failures of IT Projects as well as salient issues in the CHAOS report, and a survey was conducted between July 2003 and November 2003. The questions included were also obtained following the views of experienced software developers and project managers. The target group for the first questionnaire was software development companies. The questionnaire was aimed at gathering information about the respondents' awareness regarding prevailing project management methodologies in the Information Technology arena. In addition, information was also gathered on the tools and techniques in use as well as the level of success of projects within organisations with respect to the project completion time, cost and quality.

Another questionnaire targeted users of software products that were developed in Mauritius. Its objective was to investigate user satisfaction with respect to software developed and supplied by software companies in Mauritius. A total of 60 users was contacted and all of them responded positively to the request for information. Another objective of the questionnaire was to confirm whether or not software companies were making the necessary efforts to ensure user satisfaction with respect to timely delivery, optimum cost and an acceptable quality standard for the product concerned.

4.2.3 Problems encountered for the survey

Initially, an attempt was made to consider as many developing countries as possible but due to the low response rate the idea was dropped. Hence, only Mauritius was considered. In the case of Mauritius, a percentage response rate of 47.6% was reached. Many project managers were contacted by phone to complete the questionnaires. It was difficult to gather information from top managers in spite of reassuring them that such information would be treated as confidential. In this thesis, no information regarding the companies concerned is being revealed for confidentiality reasons.

It was also necessary to call respondents several times to complete the questionnaires or even to obtain an appointment with them to gather the information.

4.2.4 Analysis of Survey Results

The data from the questionnaire (appendix B) targeting software companies were used to determine, among others:

- the awareness of some existing project management methodologies in use by software developers and project managers;
- tasks addressed by the methodologies in use;
- project management tools in use;
- software development approach(es) in use;
- quality management standards in use;
- information reports received from companies regarding projects suffering from failure to meet deadlines, budget overruns and user dissatisfaction;
- reasons supplied by the respondents regarding their perception of the deviation in project achievement;

The data from the questionnaire targeting users of software were used to determine, among others:

- quality of software from the users' point of view;
- causes of failure of projects to meet expected objectives according to users; and
- factors to improve user satisfaction supplied by the respondents.

4.2.5 Analysis of Software Development Companies' Survey Results

The data collected were analysed using statistics like percentages and mean scores. Graphical representations were used to enhance the interpretation visually.

The first part of the questionnaire, targeting software companies, dealt with particulars of the organisations. Survey results were screened and only those dealing with software development were selected. The second part of the questionnaire comprised questions with emphasis on software project management.

The analysis of the data collected in the questionnaire is discussed next.

- **Awareness of Project Management Methodologies/Models/Standards**

It was found that many companies (60%) were not using any formal methodologies/models/standards. Information on the awareness of some popular project management methodologies/models/standards was sought from the respondents. The result (figure 4.2) showed that the majority of the methodologies/models/standards scored less than 50% regarding their awareness among software developers. The most popular methodology/model/standard being PRINCE 2®, while Euromethod and BS6079 were the least known among software developers in Mauritius.

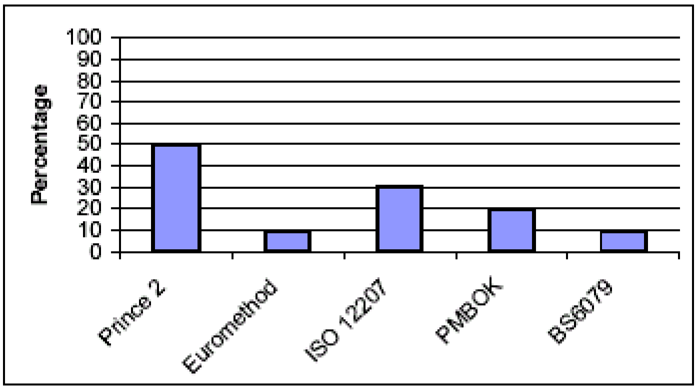


Figure 4.2: Project Management Methodologies/Models/Standards awareness

- **Tasks Addressed by Methodology/Models/Standards in use in Organisations**

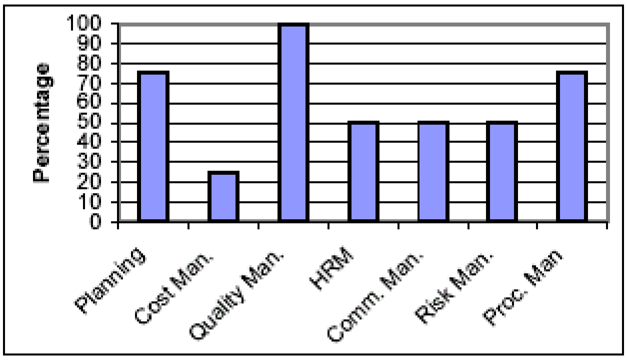


Figure 4.3: Tasks addressed by Methodology/Model/Standard

Those companies that claimed to be using a formal methodology/model/standard showed that quality management was well addressed (percentage = 100%), while cost management scored 25%, that is, the lowest percentage. This could be due to the fact that cost management was addressed separately and not included in the methodology/model/standard. The greater concern for quality management could originate from the request of clients.

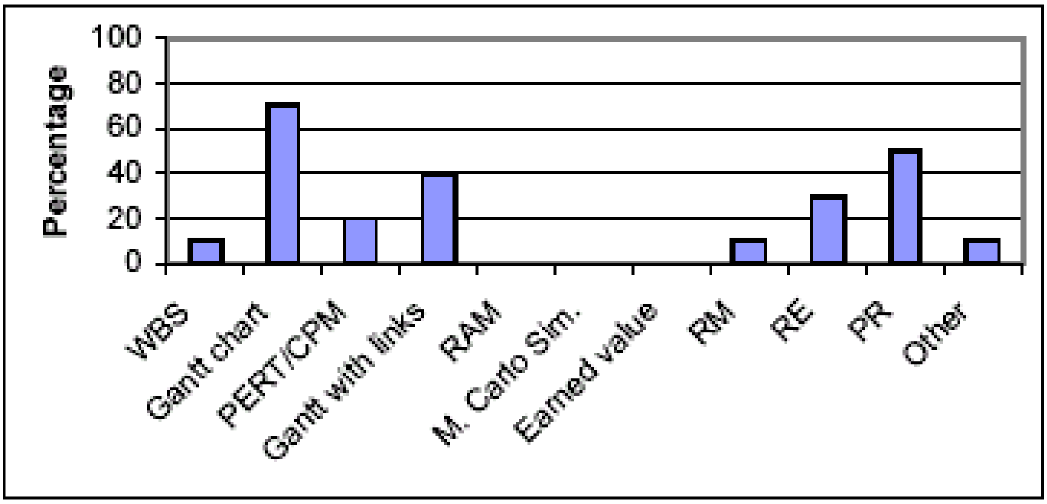
Planning and procurement management were also well addressed at around 75%. Human resource management, communications management and risk management scored 50%. Human resource management, communications management and risk management could be addressed separately rather than by the methodology/model/standard employed.

- **Support for multi-project management**

The number of projects being handled at any point may affect the successful delivery of such projects. Respondents were requested to submit information on the number of projects being managed at any one time. Around 3 projects were noted in most cases, even when no methodologies/models/standards were in use.

However, the highest number of projects (between 5 and 8) were noted in cases where a formal methodology/model/standard was used. This could be explained by the fact that a formal methodology/model/standard assisted project managers in handling more projects.

- **Project Management Tools Used**



Legend:	WBS - Work Breakdown Structure
PERT/CPM	Project Evaluation and Review Technique/Critical Path Method
RAM	Risk Assessment Matrix
M. Carlo Sim.	Monte Carlo simulation
RM	Responsibility Matrix
RE	Resource Estimate
PR	Progress Report

Figure 4.4: Project Management Tools used

It was observed that 50% of organisations were not using any project management tools although such tools may enhance the visibility of projects. Among organisations that used at least one project management tool, it was found that the Gantt chart was the most popular (scoring around 70%) followed by progress reports (around 50%). Monte Carlo simulation, risk assessment matrix and earned valued analysis were not popular among software developers. It appeared that in Mauritius, software developers chose project management tools that did not take much time to learn and implement. This could possibly be because of shortage of staff and the difficulty to allocate sufficient time for the use of software project management tools.

- **Software Development Approach**

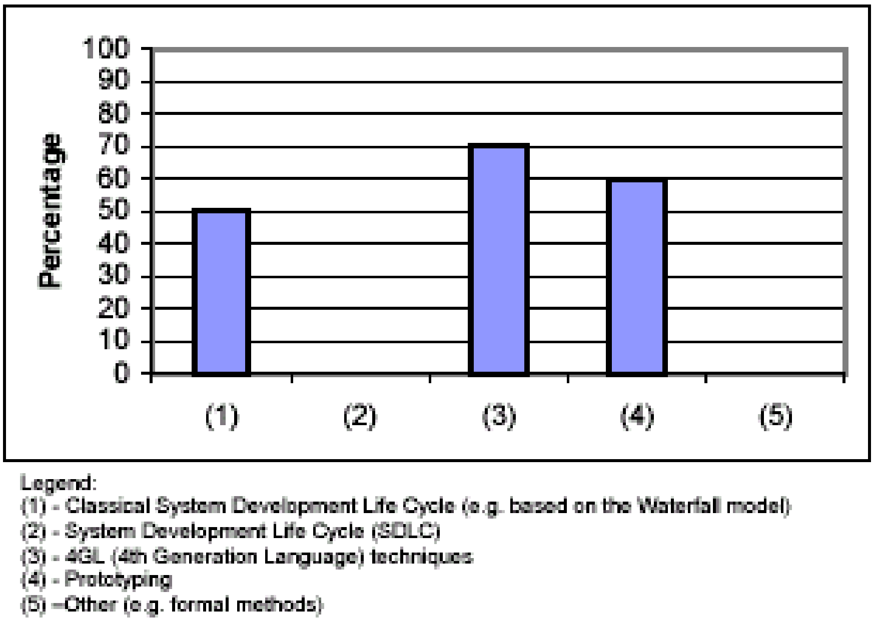
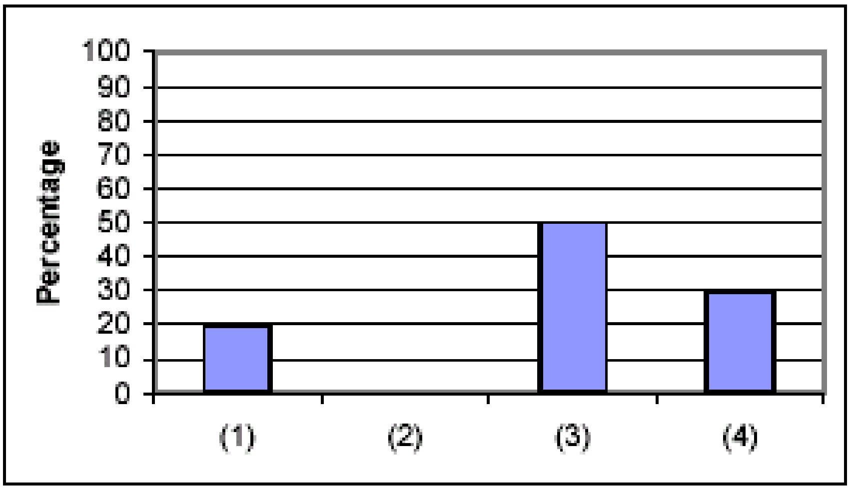


Figure 4.5: Software development approach in use

In addition to a software project management methodology/model/standard, a software development approach is required. It was found that respondents preferred relatively short development cycles. Therefore, 4GL techniques were the best preferred (70%) followed by prototyping (60%). The Classical System Development Life Cycle scored a percentage of around 50%. Software with shorter development cycles, such as prototyping, have the added advantage that users can be given a "sense" of the final product at an early stage of software development.

No respondents reported on the use of system development life cycle or formal methods. Software developers and companies were generally concerned with the development of software within the minimum period of time.

- **Quality Management Standards**



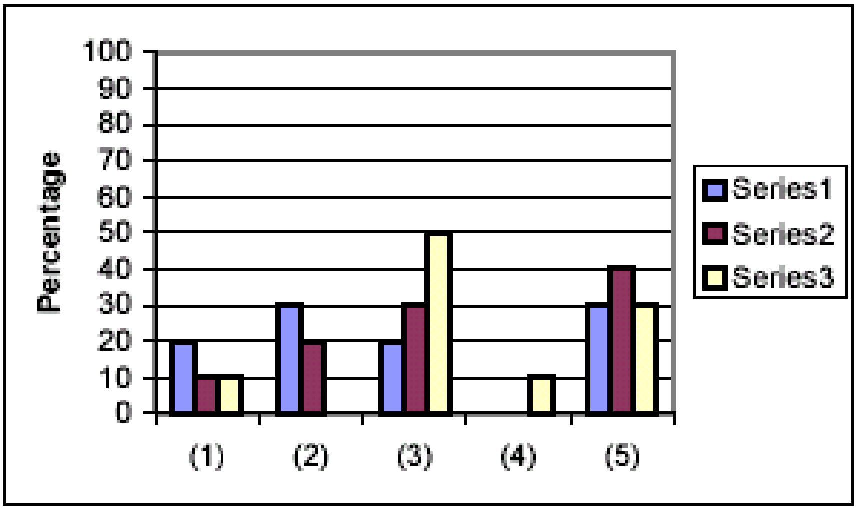
Legend:
(1) - International Standards Organisation (ISO)
(2) - Total Quality Management (TQM)
(3) - A standard developed in-house
(4) - No standard is used

Figure 4.6: Standards used

As per figure 4.6, only 20% of respondents claimed to be using an international standard, namely ISO. A total of 50% of software companies in Mauritius preferred to develop and use their own standards. This was possibly due to a lack of expertise with regards to international standards and difficulty to allocate sufficient time and resources to investigate these standards given the scarcity of qualified staff.

However, to extend their trade to other countries, an international standardised approach would certainly present Mauritian companies with a competitive advantage.

- **Projects Suffering from Failure to Meet Deadlines, Budget Overrun and User Satisfaction**



Series 1 represents projects suffering from failure to meet the deadline.
Series 2 represents projects suffering from a budget overrun.
Series 3 represents projects suffering from failure to satisfy user requirements

Figure 4.7: Projects affected by deadlines, budget and failure to meet user satisfaction

Respondents were also requested to submit information on the percentage of projects that failed to be delivered on time and within budget and failed to satisfy users annually.

As far as the failure of projects to meet deadlines was concerned:

- 20% of companies sampled claimed that more than 75% of their projects failed annually to meet the set deadline.
- 30% of companies sampled claimed that between 50% and 75% of their projects failed annually to meet the set deadline.
- 20% of companies sampled claimed that less than 50% of their projects failed annually to meet the set deadline.
- No sampled companies claimed that they did not have any project that failed annually to meet the set deadline.

In figure 4.7,

(1) - above 75%

(2) - between 50% and 75% (both inclusive)

(3) - above 0% but less that 50%
(4) - 0%
(5) - No response

The figures above indicate that improvement should be brought about in this area. Therefore, better planning could be effected and closer monitoring of projects ought to be carried out.

Regarding budget overrun:

- 10% of companies sampled claimed that more than 75% of their projects suffered annually from budget overrun.
- 20% of companies sampled claimed that between 50% and 75% of their projects suffered annually from budget overrun.
- 30% of companies sampled claimed that less than 50% of their projects suffered annually from budget overrun.
- No sampled companies claimed that they did not have any project that suffered annually from budget overrun.

A total of 40% of companies sampled did not respond to the question.

Again, much improvement would be required in this area to improve the success rate of projects.

Regarding projects suffering from failure to satisfy user requirements, the following could be noted:

- 10% of companies sampled claimed that more than 75% of their projects failed to satisfy user expectations.
- none of the sampled companies claimed that between 50% and 75% of their projects failed to satisfy user expectations.
- 50% of companies sampled claimed that less than 50% of their projects failed to satisfy user expectations.

- 10% of companies sampled claimed that none of their projects failed to satisfy user expectations.
- 30% of companies sampled did not respond to this question.

The figures above indicate that further research is required to explore the problems related to improving the success rate of software projects in Mauritius.

- **Factors responsible for Deviation in Project Achievement**

Respondents were also requested to mention about factors that were in their opinion responsible for deviation in project achievement. The results were interpreted using a Likert scale as shown below:

Strongly Agree (SA)	Agree (A)	Neutral (N)	Disagree (D)	Strongly Disagree (SD)
1	2	3	4	5

Table 4.1: Likert scale

The mean score for each question was computed. A value varying between approximately 2 and 2.5 for each of the factors was noted, which implied that respondents were of the perceived opinion that all of these factors had to be overcome for successful software projects delivery in Mauritius. The results were as follows:

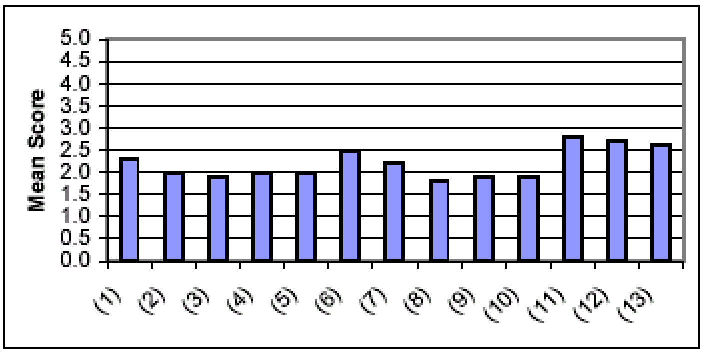


Figure 4.8: Reasons for deviation

In figure 4.8, the reasons for deviation are:

(1) - No formal methodology is used.

(2) - Poor planning.

(3) - User resistance.

(4) - Inadequate or improper communication infrastructure (between project team and customer).

(5) - Inappropriate project team.

(6) - No executive support.

(7) - Failure of project team to understand user needs.

(8) - Inadequate or absence of user involvement.

(9) - Inadequate Risk Management.

(10) - Difficulty to cope with concurrent projects.

(11) - Lack of expertise in innovative/new technology.

(12) - Lack of skilled professionals.

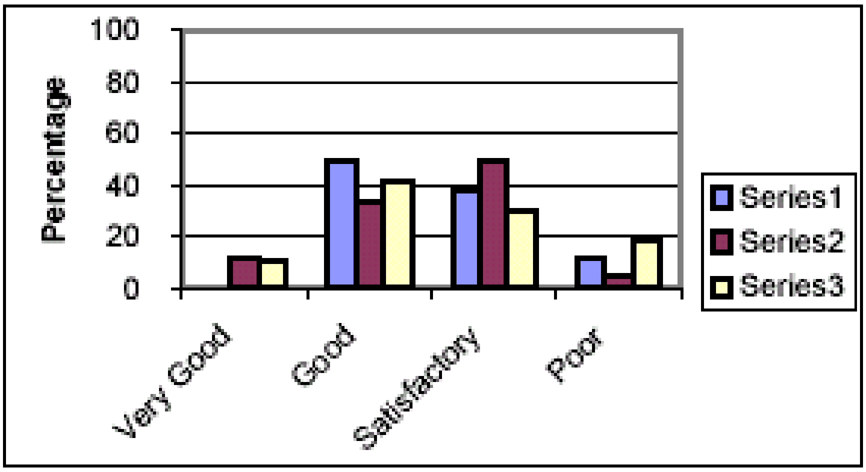
(13) - Mobility of labour.

4.2.6 Analysis of Client Survey Results

In addition to the questions targeting software development companies, it was also speculated that users of software would provide information in terms of the quality of software and reasons contributing to failure in software projects. The responses could be used to analyse any inconsistencies regarding information provided by software companies.

The data collected was analysed and represented graphically in terms of percentages and mean scores.

- **Quality of Software from a User Perspective**



Series 1 represents user friendliness of software.
Series 2 represents performance of software.
Series 3 represents security features of software.

Figure 4.9: Rating of quality

The majority of users of software were satisfied with the user friendliness features of software developed locally (50% of respondents view this feature as "Good" and 40% as "Satisfactory"). This could be attributed to the software development techniques used in Mauritius, whereby users were exposed early in the development phase to a prototype version of the software under development.

A total of 50% of the respondents were satisfied with the performance of the said software. Respectively, 10% and 35% of users view software as "very good" and "good" in terms of performance. Again, this could be attributed to the short development cycles, whereby users were exposed early to the software under development.

A total of 10% of the respondents rated security features as "Very Good", 40% of users as "Good" and 30% as "Satisfactory". Around 20% of users were of the opinion that security features were poorly addressed. The high degree of user satisfaction in this respect could also be attributed to the short development cycles, whereby users were exposed early to the software under development.

- **Causes of Project Failure to Meet Expected Objectives from a User Perspective**

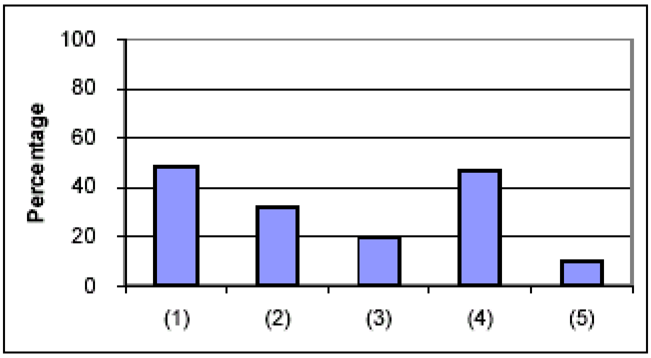


Figure 4.10: Causes of failure

In figure 4.10 above the causes of project failure to meet the expected user objectives were:

(1) - Lack of communication between supplier and customer.
(2) - Lack of experience on the part of software developer.
(3) - Unprofessional mind-set of supplier.
(4) - Possible lack of best practices for software development and IT management.
(5) - None of the above reasons are applicable (i.e. other reasons exist).

Most users were of the opinion that the causes of failure to meet expected user objectives could be attributed to “lack of communication between supplier and customer” and “possible lack of best practices for software development and IT management”.

It was, therefore, noted that an improvement in communication and the need to review software development practices by the developers as well as better IT management could improve the successful delivery of software projects.

- **Factors to Improve User Satisfaction from a User Perspective**

The following factors were considered to evaluate customer satisfaction based on a questionnaire (appendix B):

(1) - Improving communication between suppliers and users through regular meetings or informal discussions.
(2) - Frequent participation of users in the software development process by providing regular feedback.
(3) - Suppliers should consider user satisfaction seriously rather than just abiding strictly by the specifications agreed upon initially on award of a contract.
(4) - A Chief Information Officer (CIO) in charge of issues pertaining to IT should be appointed on the user side.
(5) - Each user department must recruit its own software developers, IT manager, System Analysts, Database Administrators, etc.
(6) - More effort should be devoted to the elimination of errors in software.

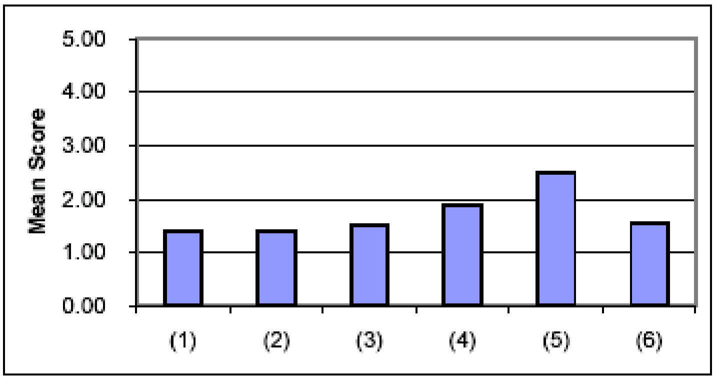


Figure 4.11: Factors to improve user satisfaction

According to figure 4.11, users are of the opinion that all the factors above are important, except for the recruitment of software developers for each department where a neutral opinion was provided.

The responses were analysed on a Likert scale. It was observed that the mean score of the responses lied mostly in the "Agree" to "Strongly Agree" category (mean score of between 1.5 and 2) for most of the factors, except for the factor

(5). Therefore, from a user point of view, most of the factors considered had to be addressed during project management.

4.3 Status of software project management in the public sector

In addition to the results of the survey carried out by Sukhoo et al. (2004b), Boodhun (2005) also investigated the causes of the failure of IT projects in the Mauritian Public Sector. A case study of 10 software projects was considered. Interviews were carried out to gather in-depth information from the respondents.

The research revealed the following main problems:

- Lack of software project management methodology.
- Shortcomings in time management. The developers were involved in multiple projects and could not manage their time properly.
- Lack of commitment, planning and coordination at the level of users.
- Lack of planning of activities on the part of the project team.
- Lack of communication between project team and users.
- Lack of expertise with the development platform (in some cases).
- User requirements were not well defined.
- Users were not satisfied with the final product (in some cases). A prototype could have solved this problem.
- No quality checks were performed (in some cases).
- Lack of risk analysis and management.
- An average delay of 6 months was encountered and no project was delivered on time.

Boodhun (2005) concluded that the adoption of best practices of software project management would certainly help the Mauritian Government software development department to attain its goal in delivering efficient systems while respecting target dates. In addition, this initiative would strengthen the good working relationships with the client Ministries and also boost up the work morale of the staff.

4.4 Conclusion

The survey conducted by Sukhoo et al. (2004b) targeted two categories of respondents, namely software development companies and users of locally developed software. The following points could be noted:

- A lack of formal methodologies was used.
- An insufficient number of companies was using project management tools.
- An improvement in quality of software could be achieved by adopting standards like ISO or TQM.
- More consideration should be given to the factors causing deviation in project achievement.
- An improvement in communication between software developers and users was required.
- The need for best practices in the field of software development.

The result of this survey provides an insight into the status of software project management in the country. It could be noted that improvement was required so as to allow software companies to deliver software products to contribute towards the vision set by the Government to transform Mauritius into a Cyber Island. The ICT sector should evolve rapidly to extend the trade of software with other countries and consequently meet the objective of developing the fifth pillar of the economy.

The survey conducted by Boodhun (2005) also confirmed several areas where project management problems are encountered and where improvements are envisaged. However, the research was aimed at the public sector where many of the software products are developed for various Ministries and Government departments.

It is, however, believed that further information needs to be gathered in order to capture the status of software project management in Mauritius. Therefore, the results of another survey are being considered in chapter 6 to decide on the way forward regarding software project management in Mauritius.

In chapter 5, a basis for the survey of maturity level of organisations considered in chapter 6 is provided. Several project management maturity models that are currently available are, therefore, discussed in chapter 5. These models are different from the so-called sequential models as the organisations can be classified into different levels of maturity. The progression from one level to the next higher level depends on the focus areas successfully achieved. The survey, presented in chapter 6, is important as it has been established that there is a correlation between project management maturity and project success by Sonnekus and Labuschagne (2004) as well as Schiltz (2003).

Chapter 5

Current Project Management Maturity Models

Table of contents

5.0 Chapter layout

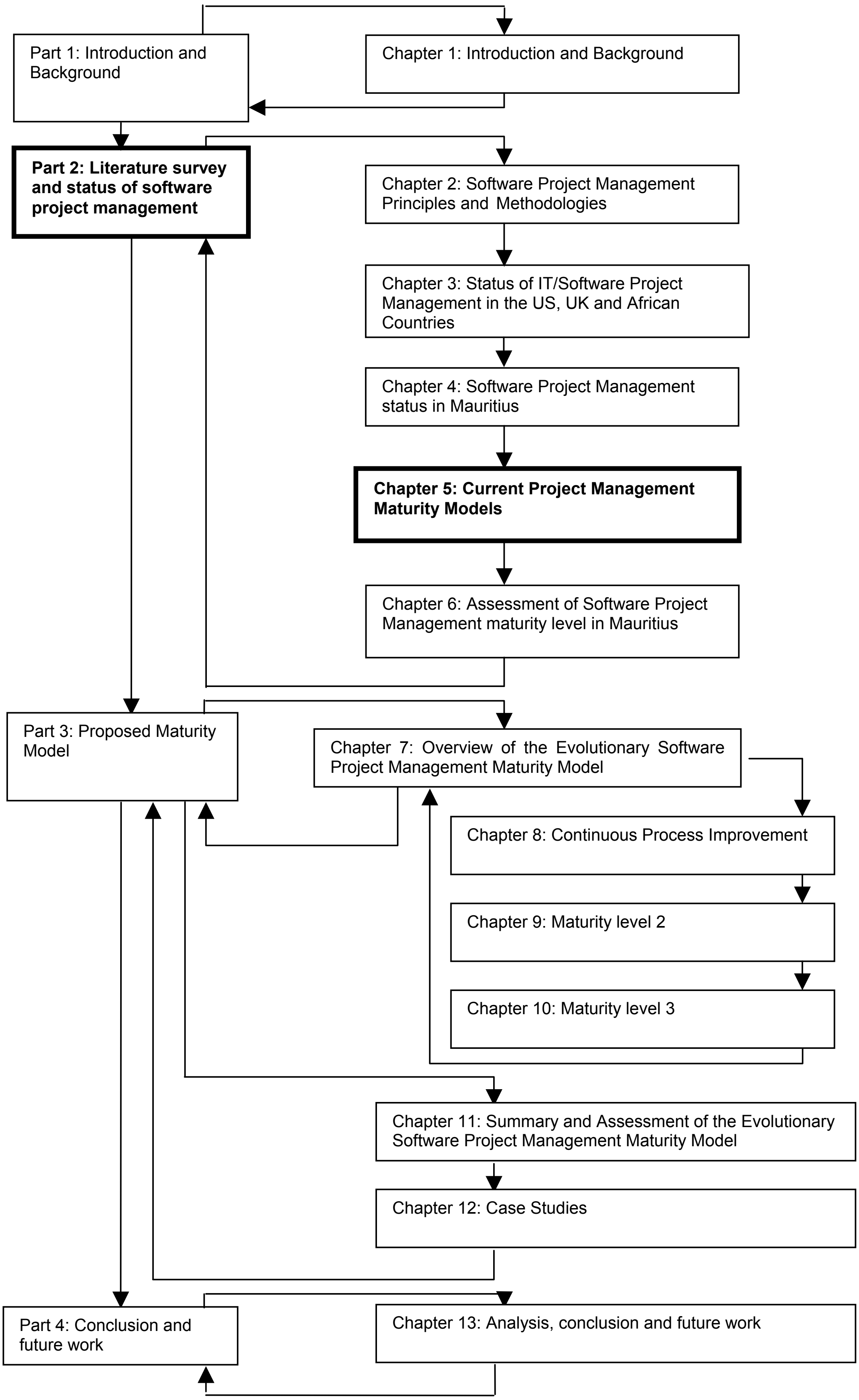


Figure 5.1: Chapter 5 in context within the overall thesis

5.1 Introduction

Some organisations manage their projects using maturity models. Such models have a number of maturity levels (typically 5 levels are defined), each one comprising of some process areas (or focus areas) that are necessary for the management of projects. A definition of maturity model is provided in the next subsection.

Project management is supported by project management methodologies/models/standards. Organisations achieve progressive development of their project management approach, using both qualitative and quantitative data to make decisions. Just as individuals benefit from achieving personal maturity, organisations can benefit from achieving organisational project management maturity (Project Management Institute, 2004b).

Various Project Management Maturity Models (PMMMs) have been developed (e.g. MicroFrame self-assessment tools; Project Management Process Maturity $(PM)^2$; Kerzner's maturity model; the Software Engineering Institute's (SEI's) Capability Maturity Model Integration; PRINCE 2 Maturity Model; OPM3^{TM}). Most of them are defined by five maturity levels, namely 1 through 5.

Project Management Maturity Models can be categorised as generic models and software specific models as shown in figure 5.2 below:

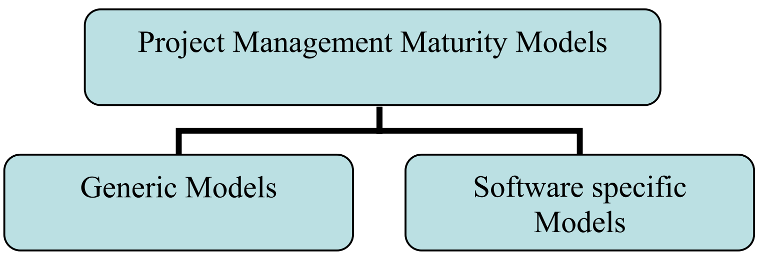


Figure 5.2: Categories of project management maturity models

Generic models are used for all types of projects while the software specific models are used for the management of software projects. Software specific models take care of specificities inherent in software projects. The greater part of this chapter deals with discussions of these models.

5.1.1 Aim, objectives and structure of this chapter

The aim of this chapter is to present a literature review of current project management maturity models. According to Cooke-Davis (2002:4), "As project management matures as a business discipline, we will inevitably face a greater demand to demonstrate the maturity of the processes that we advocate". Most of the models are for general applications, i.e. for both IT and software projects as well as traditional engineering projects. For the relevance of this thesis, both types have been considered but with the specific focus on software project management.

According to the Oxford Advanced Learner's dictionary (Wehmeier, 2005), maturity is defined as follows:

Maturity is the state of being fully grown or developed …………...Definition 5.1

According to the Project Management Institute (2004a), a maturity model is defined as:

A Maturity model is a conceptual framework with constituent parts that defines maturity in the area of interest …………...Definition 5.2

A project management maturity model comprises areas of interests arranged in maturity levels. A higher maturity level comprises an area of interest that is more mature than a lower maturity level.

The objective of this chapter is to provide a foundation for the development of a maturity model for software project management in Mauritius as an instance of developing countries.

The remainder of the chapter is structured as follows:

Section 5.2 – In this section, the Berkeley Project Management Process Maturity Model is described. It is a model with 5 maturity levels.

Section 5.3 – This section comprises the description of the Microframe's Self Assessment Tool. Five maturity levels are available in the model.

Section 5.4 – The PRINCE 2 Maturity Model is described in section 5.4. PRINCE 2® also has a sequential model which was described in chapter 2. In PRINCE 2 maturity model, only 3 maturity levels are defined as compared to most other models.

Section 5.5 – The Organizational Project Management Maturity Model is discussed in section 5.5. This is a framework that was released by the Project Management Institute (PMI). The PMI also has a sequential model, the PMBOK®, that was described in chapter 2.

Section 5.6 – The Kerzner's Management Maturity Model is a general management framework with 5 maturity levels. In this section, the model is discussed in terms of the requirements to achieve each of the different maturity levels.

Section 5.7 – In this section, the Software Engineering Institute's (SEI) Capability Maturity Model Integration is described. This model has 5 maturity levels, each one with a number of key process areas (or focus areas). This model has been specifically developed for software project management and organisations that have achieved the highest level have been very successful (Software Engineering Institute, 2005).

Section 5.8 – The ISO/IEC Software Process Assessment, described in this section, has six maturity levels.

Section 5.9 – The concluding remarks in this section highlight and motivate the need for a software project management of an indigenous nature for developing countries. It is believed that a framework that embodies the social, cultural, economic, political and soft skills can show better acceptance from organisations in these countries. In addition, given the correlation between maturity level and success rate of projects, it is speculated that a maturity-based model may be suitable for developing countries.

5.2 Berkeley Project Management Process Maturity Model $(PM)^2$

Kwak and Ibbs (2002) proposed a 5-level project management maturity $(PM)^2$ model to assess and improve an organisation's maturity level. The model is shown in figure 5.3. Each level divides project management processes and practices into nine knowledge areas (Integration, Scope, Time, Cost, Quality, Human Resource, Communications, Risk and Procurement) and five phases (Initiate, Plan, Execute, Control and Close out) as shown in figure 5.4, thus adopting the classification of the Project Management Body of Knowledge (Kwak and Ibbs, 2002).

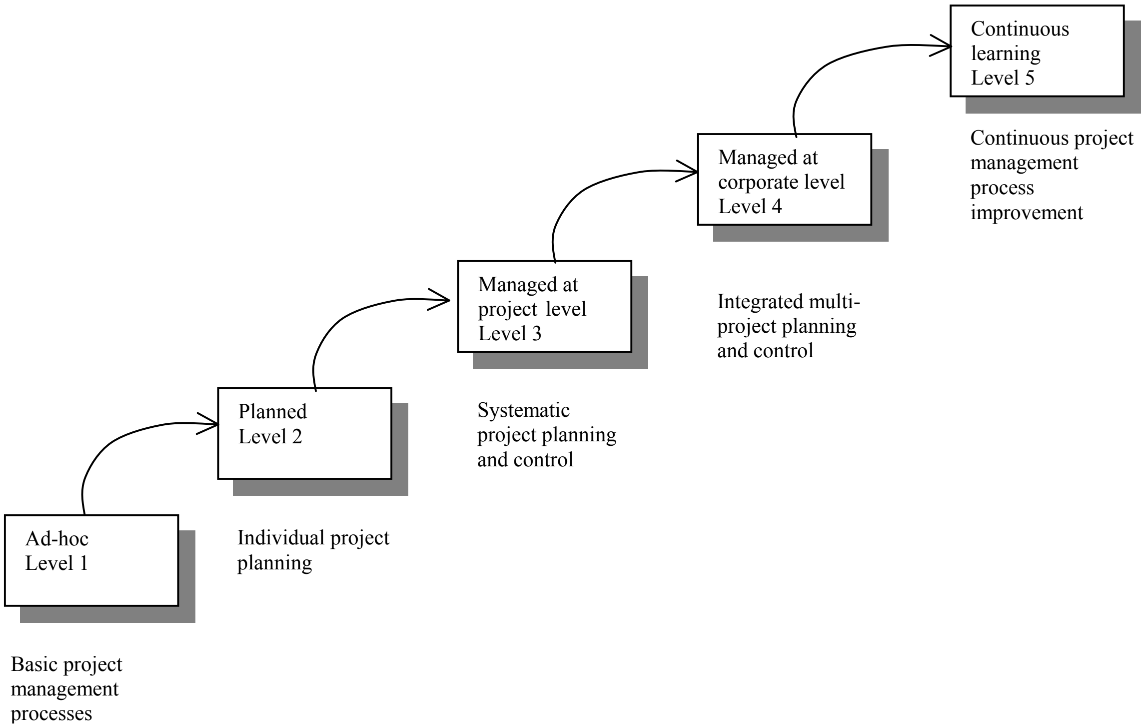


Figure 5.3: Project management process maturity model (Kwak and Ibbs, 2002)

Each of the levels is discussed next.

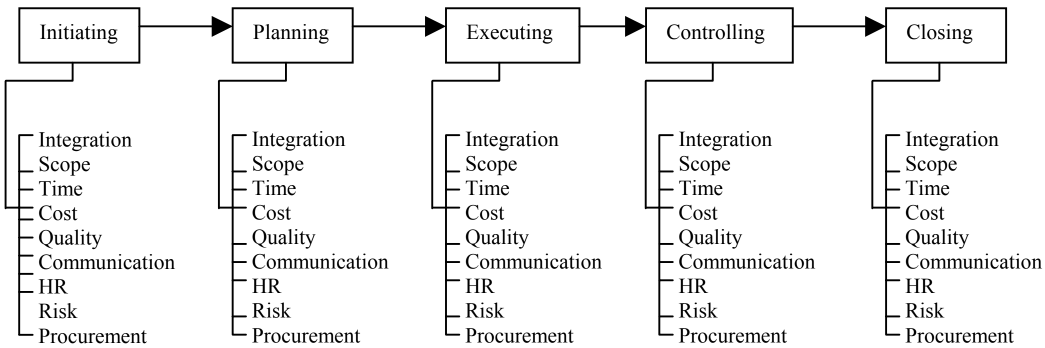


Figure 5.4: Integrating project processes and project management knowledge areas (Kwak and Ibbs, 2002)

Level 1 – Ad-hoc stage

At this level, no project management processes or practices are consistently available. The execution of a project depends on the project team experience.

Level 2 – Planned stage

Informal processes are defined and informal problems are identified at this level.

Level 3 – Managed stage

Formal project planning and control systems are managed at level 3.

Level 4 – Integrated stage

Multiple projects are managed efficiently and data regarding processes are quantitatively analysed, measured and stored.

Level 5 – Sustained stage

Project management processes are fully understood and continuously improved.

The **shortcomings** of the Berkeley project management process maturity model are:

(i) Political, socio-economic and cultural aspects are not considered.
(ii) Soft skills are not considered.
(iii) The model is a generic one and, therefore, it does not address specific problems inherent in software project management.

5.3 Microframe's Self Assessment Tool

This project management maturity model was developed by Micro Frame Technologies, Inc. and Project Management Technologies, Inc. (Enterprise Planning Associates, 1998). A self-assessment tool comprising 50 multiple choice questions was also designed to assess the maturity level of an organisation.

According to Enterprise Planning Associates (1998), the model is a process and system framework for helping organisations to improve their project management practices and management of project functions. It provides a phased set of maturity descriptions, improvement criteria, operating metrics, and questions that can be used to assess the current level of maturity and develop a focused plan for improving the effectiveness of project and functional management. The objective of the model is to improve the ability of organisations to consistently achieve goals for cost, schedule and customer satisfaction.

The model emphasises incremental improvement in single and multi-project management disciplines and ranks maturity across five levels, as follows:

Level 1 - Ad-Hoc

The project management process is described as disorganised, and occasionally even chaotic. Systems and processes are not defined. Project success depends on individual effort. Projects are prone to chronic cost and schedule problems.

Level 2 - Abbreviated

Some project management processes and systems are established to track cost, schedule, and performance. Underlying disciplines, however, are not well understood or consistently followed. Project success is largely unpredictable and cost and schedule problems are the norm.

Level 3 - Organized

Project management processes and systems are documented, standardised, and integrated into an end-to-end process for the company. Project success is more predictable. Cost and schedule performance is improved.

Level 4 - Managed

Detailed measures of the effectiveness of project management are collected and used by management. The process is understood and controlled. Project success is more uniform. Cost and schedule performances conform to a plan.

Level 5 - Adaptive

Continuous improvement of the project management process is enabled by feedback from the process and from piloting innovative ideas and technologies. Project success is the norm. Cost and schedule performances are continuously improving.

The **shortcomings** of the Microframe self-assessment tool are:

(i) Political, socio-economic and cultural aspects are not considered.
(ii) Soft skills are not considered.
(iii) The model is a generic one and, therefore, it does not address specific problems inherent in software project management.

5.4 PRINCE 2 Maturity Model

According to the Office of Government Commerce (2004), the PRINCE 2 Maturity Model has the purpose of enabling organisations to gauge, by assessment, their maturity in the use of the model. Each maturity level has "*Key Process Areas*" and "*Key Practices*". The latter describes what an organisation has to do to manage projects effectively. The former are the focus areas on which the organisation has to concentrate so as to improve its success rate of delivery of software projects. The PRINCE 2 Maturity Model assesses maturity of organisations at three levels as follows:

Level 1 - Initial

At this level, the organisation recognises project. No key process areas are defined at the initial level.

Level 2 - Repeatable

At this level, the model is used across the organisation but inconsistently. Eleven (11) key process areas can be identified at the repeatable level and they are:

(i) Directing a project.
(ii) Initiating a project.
(iii) Controlling a stage.
(iv) Closing a project.
(v) Business case.
(vi) Organisation.
(vii) Plans/planning.
(viii) Management of Risk.
(ix) Controls.
(x) Quality management.
(xi) Configuration management and change control.

Level 3 – Defined

PRINCE 2® is applied consistently across the organisation. The following five key process areas (KPAs) are defined at this level:

(i) Organisational focus.
(ii) Tailoring of PRINCE 2®.
(iii) Training in PRINCE 2®.
(iv) Integrated management.
(v) Quality assurance.

The **shortcomings** of the PRINCE 2 maturity model are:

(i) Political, socio-economic and cultural aspects are not considered.

(ii) Soft skills are not considered.

(iii) Given that PRINCE 2 maturity model evolved from PRINCE 2®, people management and contract management are not covered (12Manage, 2006).

(iv) The model is a generic one and, therefore, it does not address specific problems inherent in software project management.

5.5 Organizational Project Management Maturity Model (OPM3™)

The Project Management Institute (2004a) defines organisational project management as the application of knowledge, skills, tools and techniques to organisational and project activities. It aims to achieve the goals of an organisation through projects or the systematic management of projects, programs and portfolios in alignment with the achievement of strategic goals. OPM3™ is a maturity model developed by the Project Management Institute for organisations to assess their maturity as well as to increase their maturity when applicable. The model is aligned with the PMBOK®. As shown in figure 5.5, it consists of one of the two dimensions with a view in terms of their association with the progressive stages of process improvement – from standardisation through measurement to control and finally to continuous improvement (Project Management Institute, 2004a). Another dimension involves the view of addressing Project Management, then Program Management and finally Portfolio Management (Project Management Institute, 2004a).

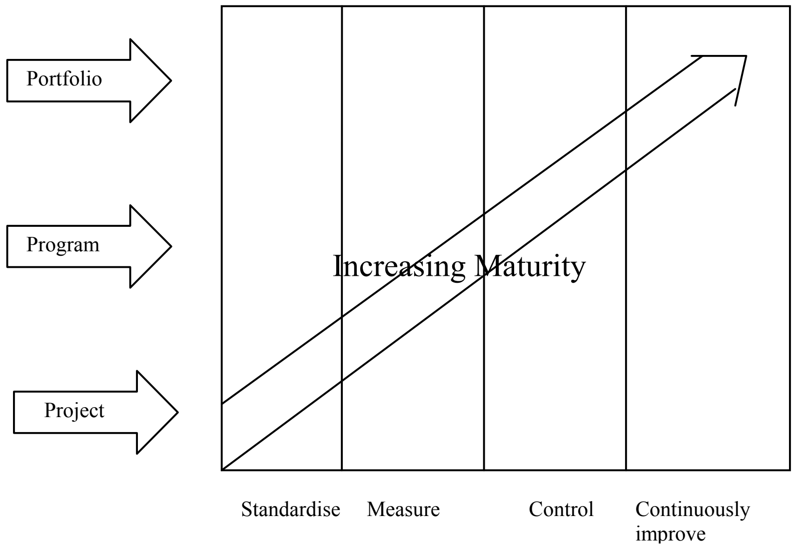


Figure 5.5: Organizational Project Management Maturity Model (Project Management Institute, 2004a)

OPM3™ has three interlocking elements, namely Knowledge, Assessment and Improvement. The Knowledge element provides the body of best practices for users. The Assessment element allows for a comparison of the organisation's project management practice against the best practices in OPM3™. The Improvement element allows organisations to make use of the assessment results to move ahead with change initiatives. These elements are applied in a cycle called the OPM3™ cycle as shown in figure 5.6.

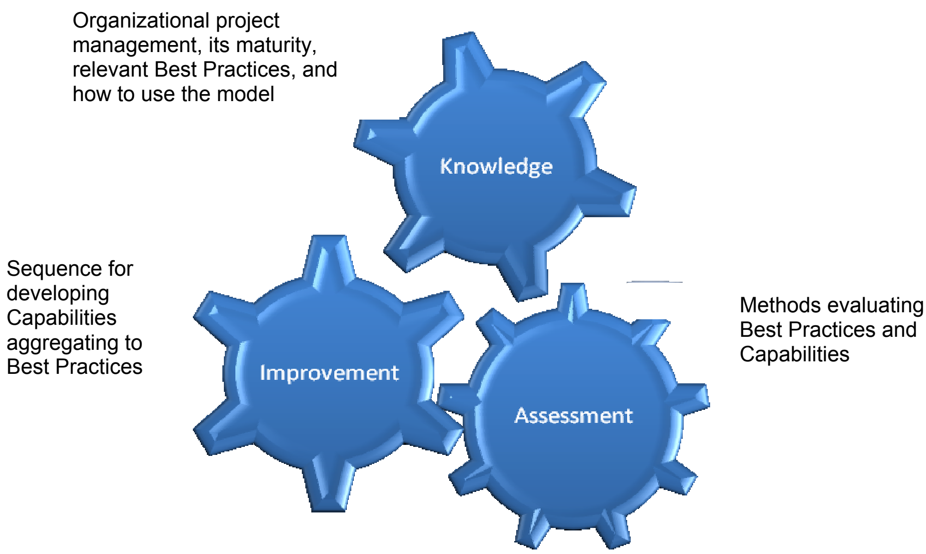


Figure 5.6: OPM3™ cycle (Project Management Institute, 2004a)

The **shortcomings** of the OPM3™ are:

(i) Political, socio-economic and cultural aspects of developing countries are not taken into consideration.
(ii) Soft skills are not considered.
(iii) The model is a generic one and, therefore, it does not address specific problems inherent in software project management.

5.6 Kerzner's Management Maturity Model

Dr. Harold Kerzner developed a generic project management maturity model comprising 5 levels. Project management is regarded as a core competency that many companies must develop in order to remain competitive in the market (Schiltz, 2003). The various levels are depicted in figure 5.7 (International Institute for Learning, 2002).

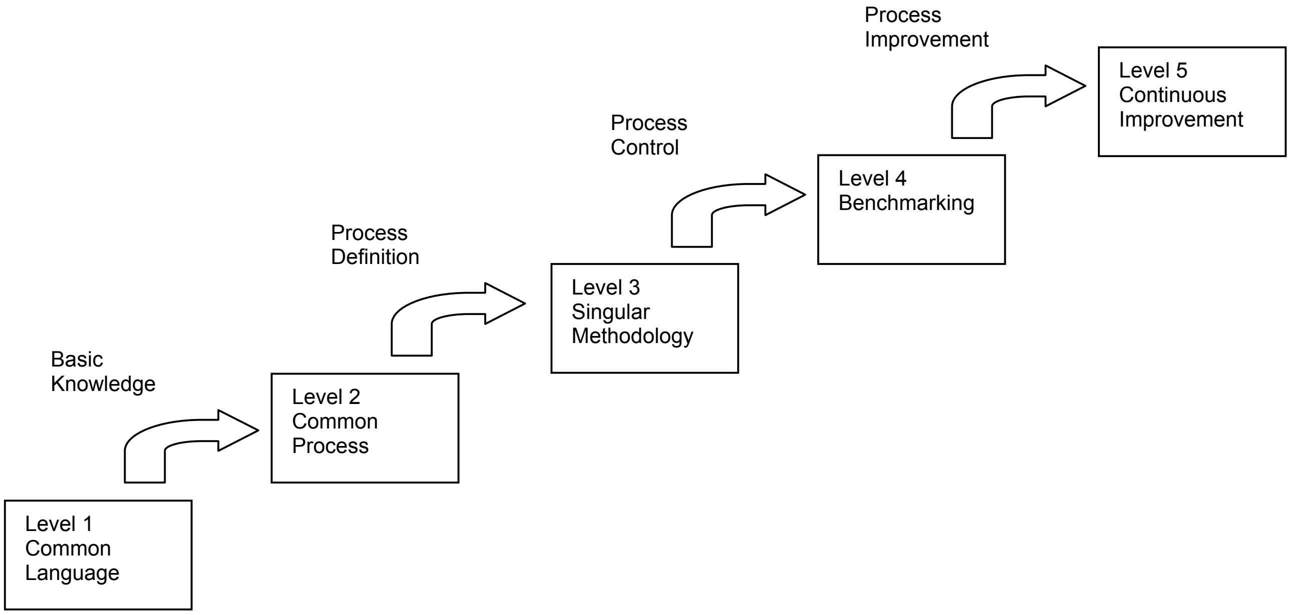


Figure 5.7: Kerzner's project maturity levels (adapted from Harrison, Sweeney, Taylor and Wood, 2003)

According to Harrison et al. (2003), the different levels are described as follows:

Level 1 – Common Language

Common Language is the basic knowledge of PM and the terminology used. Eight management categories are defined at this level and they are:

- Scope/Integration.
- Time.
- Cost.
- Human Resources.
- Procurement.
- Quality.
- Risk.
- Communication.

Level 2 – Common Process

The Common Processes that are defined and developed are applicable and repeatable. Assessment for level 2 is divided into five life cycle phases:

- Embryonic - basic recognition of the need for and benefits of PM.
- Executive Management Acceptance - executive support in utilising PM.
- Line Management Acceptance - line/functional management support in utilising PM.
- Growth - development of a PM methodology.
- Initial Maturity development and usage of a cost and schedule control system; ongoing PM education/training program.

Level 3 – Singular Methodology

Singular Methodology is the synergistic effect of combining all corporate methodologies. At this level, disjointed processes and methodologies are merged into a single methodology, which can be measured by the following six traits:

- Integrated Processes - multiple possible processes have been combined into one.
- Cultural Support - corporate culture encourages collaborative efforts to promote Project Management.
- Management Support - each level of management supports the methodology through their roles.
- Informal Project Management - methodology supported by informal means.
- Training and Education - acknowledgement that PM training and education reaps rewards.
- Behavioural Excellence - training for PM to replace line management mentality.

Level 4 – Benchmarking

Benchmarking process improvement is required to maintain a competitive advantage. Benchmarking can be a powerful tool for assessing cycle time, quality, resource allocation, training practices, sales productivity, and other business-related issues. For a benchmarking program to succeed, it must evaluate the right metrics, measure those metrics accurately and relevantly, and report the metrics clearly in a timely fashion.

An organisation must meet the following four key level 4 requirements to advance to level 5:

- Create an organisation dedicated to benchmarking.
- Develop a project management benchmarking process.
- Decide what to benchmark and against whom to benchmark.
- Recognise the benefits of benchmarking.

Level 5 – Continuous Improvement

Continuous Improvement evaluates the enhancement to Project Management from each improvement. At level 5, the organisation evaluates and analyses all the lessons learned from the previous levels and implements required changes to improve PM processes.

The **shortcomings** of the Kerzner's management maturity model are:

(i) Political, socio-economic and cultural aspects (the model restricts culture to corporate culture only) do not take into consideration the specific needs of developing countries.

(ii) Soft skills are not considered.

(iii) It is a generic methodology and does not specifically address software project management.

5.7 SEI's Capability Maturity Model Integration (CMMI)

The Capability Maturity Model Integration (CMMI) was developed by the Software Engineering Institute (Software Engineering Institute, 2005). It evolved from the Capability Maturity Model (CMM). Instead of using multiple CMMs, the CMMI product team combined three models, namely the Capability Maturity Model for Software (SW-CMM), the Electronic Industries Alliance Interim Standard (EIA/IS) and the Integrated Product Development Capability Maturity Model (IPD-CMM) into a single framework for enterprise-wide process improvement (CMMI product team, 2002). The CMMI provides a relatively easy migration from SW-CMM to CMMI.

The key components of the CMMI are process areas and these are grouped into five maturity levels (CMMI Product Team, 2002). There are 25 process areas in total. Each process area has specific and generic goals. Specific goals have specific practices. Generic goals have common features that organise generic practices. The organisation of the CMMI is shown in figure 5.8.

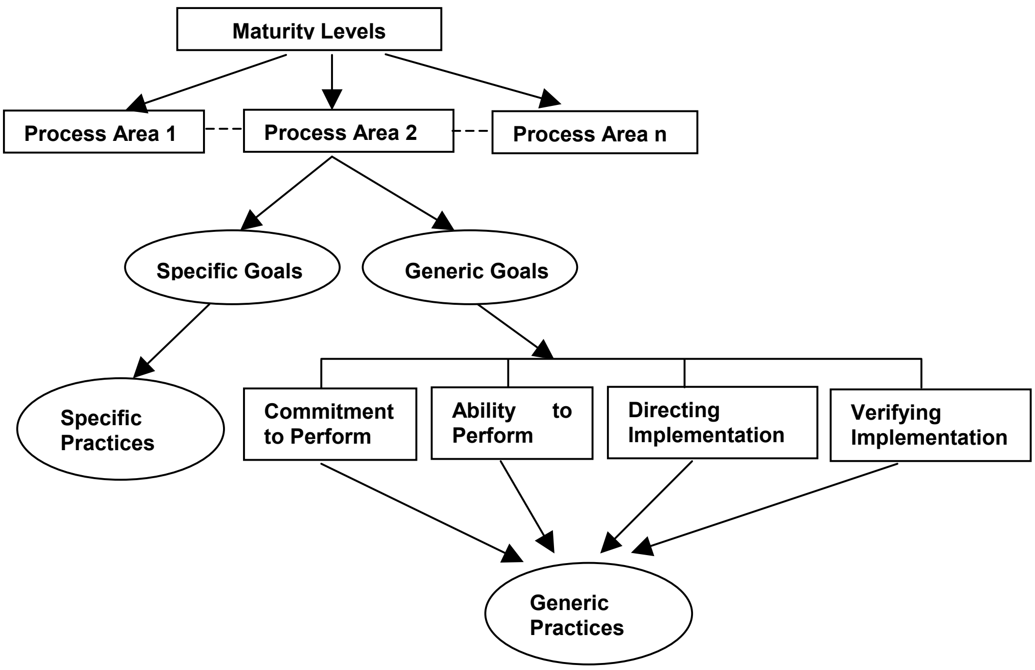


Figure 5.8: CMMI Model Components (CMMI Product Team, 2002)

The CMMI has five maturity levels as shown in figure 5.9 (Hefner, 2003). Maturity level 2 onwards has process areas that are focus areas aimed at achieving a particular maturity level.

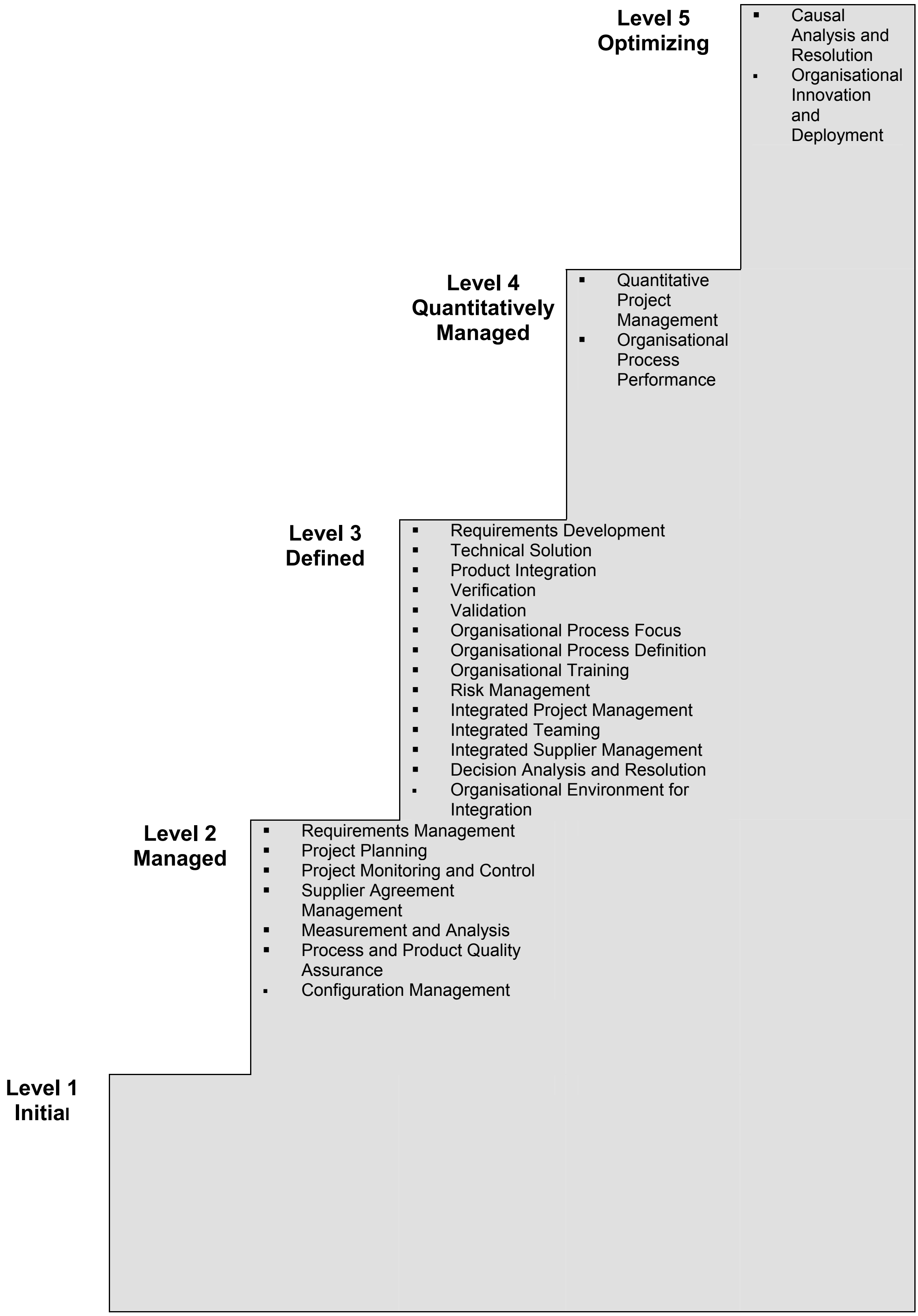


Figure 5.9: CMMI Model (Hefner, 2003)

The levels are described next.

Level 1 – Initial

At this level, processes are usually ad-hoc and chaotic. The success of projects depends largely on the skills of the project members. Although organisations at level 1 often produce products that work, they frequently exceed their budget and overrun their schedule. During a crisis, organisations tend to abandon processes and are not able to repeat past successes.

Level 2 – Managed

At this level, processes are planned, performed, measured and controlled. All the specific and generic goals may be achieved for the process areas concerned at the level. Plans are documented.

Level 3 – Defined

Processes are well characterised and understood, and are described in standards, procedures, tools and methods. All processes are consistent across the organisation except where the tailoring of guidelines is required.

Level 4 – Quantitatively Managed

At this level, quantitative objectives for quality and process performance are established. Processes are qualitatively and quantitatively predictable.

Level 5 – Optimizing

This level focuses on continually improving process performance through incremental and innovative technological improvements.

The **shortcomings** of the CMMI are:

(i) The adoption of CMM encourages so much bureaucracy and rigidity that organisations may be involved in lengthy procedures (Herbsleb et al., 1997). The CMMI does not seem to have improved on this issue.

(ii) The CMM requires the experience of very knowledgeable people to succeed (Bach, 1994). The training of staff and the help of consultants are required to implement CMMI. Investment on training for CMMI may not be possible for small or even medium sized organisations.

(iii) Political, social, economic and cultural aspects are not considered in CMMI.

(iv) The documentation on CMMI is voluminous as the guide contains 639 pages. It may take much time for an organisation to become conversant with the framework without the help of consultants.

(v) Worldwide, 139 organisations adopting CMMI are at level 5 and 30 organisations are certified at level 4. The distribution of 583 organisations using CMMI in the world is shown figure 5.10 (Software Engineering Institute, 2005). The number of organisations adopting CMMI is still low and the majority appears to be below level 4.

(vi) Soft skills are not included as a process area.

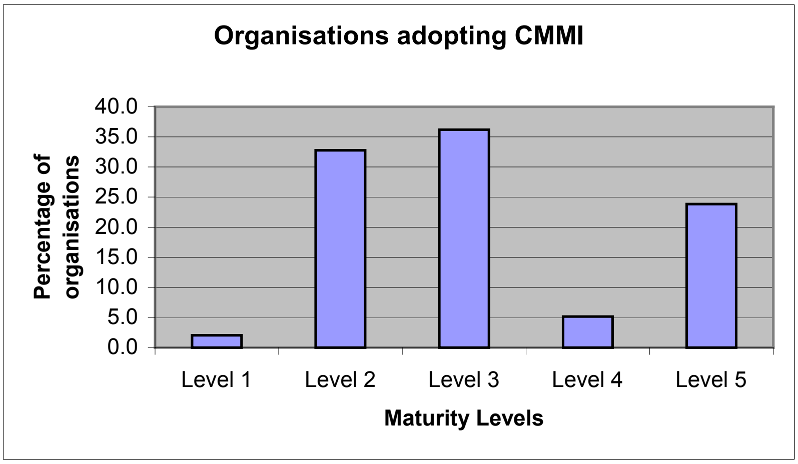


Figure 5.10: Organisations distributed by maturity levels (Software Engineering Institute, 2005)

5.8 ISO/IEC Software Process Assessment

In June 1995, the Software Process Improvement and Capability dEtermination (SPICE) Project Organisation put forward a set of working drafts among which was a model for process management (SPICE Project, 1995). The model has six maturity levels described below:

Level 0 - Not-Performed

At this level, there is general failure to perform the base practices in the process. There are no easily identifiable work products or outputs of the process.

Level 1 - Performed-Informally

Base practices of the processes are generally performed at level 1. The performance of these base practices may not be rigorously planned and tracked. Performance depends on individual knowledge and effort. Work products of the process testify to the performance. Individuals within the organisation recognise that an action should be performed, and there is general agreement that this action is performed as and when required. There are identifiable work products for the process.

Level 2 - Planned-and-Tracked

At level 2, performance of the base practices in the process is planned and tracked. Performance according to specified procedures is verified. Work products conform to specified standards and requirements.

The primary distinction from the Performed-Informally Level is that the performance of the process is planned, managed and progressing towards a well-defined process at level 3.

Level 3 - Well-Defined

Base practices are performed according to a well-defined process using approved, tailored versions of standard, documented processes. The primary distinction from the Planned-and-Tracked Level (level 2) is that the process of the Well-Defined Level is planned and managed using an organisation-wide standard process.

Level 4 - Quantitatively-Controlled

Detailed measures of performance are collected and analysed. This leads to a quantitative understanding of process capability and an improved ability to predict performance. Performance is objectively managed. The quality of work products is quantitatively known. The primary distinction from the Well-Defined Level (level 3) is that the defined process is quantitatively understood and controlled.

Level 5 - Continuously-Improving

Quantitative process effectiveness and efficiency goals (targets) for performance are established at this level, based on the business goals of the organisation. Continuous process improvement against these goals is enabled by quantitative feedback from performing the defined processes and from piloting innovative ideas and technologies.

The primary distinction from the Quantitatively-Controlled Level is that the defined process and the standard process undergo continuous refinement and improvement, based on a quantitative understanding of the impact of possible changes to these processes.

This model is rather similar to the SEI's Capability Maturity Model and has some disadvantages.

The **shortcomings** of the ISO/IEC software process assessment:

(i) Political, socio-economical and cultural aspects are not considered.

(ii) Soft skills are not considered.

5.9 Conclusion

In this chapter, various maturity models were presented. Some of them are applied to all types of projects while others have been developed for the software industry. Their shortcomings have also been highlighted.

All the project management maturity models considered above revealed shortcomings like a lack of soft skills, political, socio-economic and cultural aspects. According to Muriithi and Crawford (2003) as well as studies conducted in Mauritius (Sukhoo et al., 2004b), project management approaches of an indigenous nature (i.e. those project management approaches that are developed locally and not brought from other countries) are required in African developing countries.

However, the study of maturity models, in this chapter, left no stone unturned in the choice of a maturity model for developing countries. It can, thus, be confirmed that a software project management methodology/model/standard of an indigenous nature is needed for developing countries.

Furthermore, project management maturity models are becoming popular. The Office of Government Commerce in the UK and the Project Management Institute in the US have both released their project management maturity models. This shows that there is an increasing interest for maturity models.

In the next chapter, an assessment of the maturity level of software development organisations in Mauritius, as a developing country, is made.

Chapter 6

Assessment of software project management maturity level in Mauritius

---

Table of contents

6.0 Chapter layout

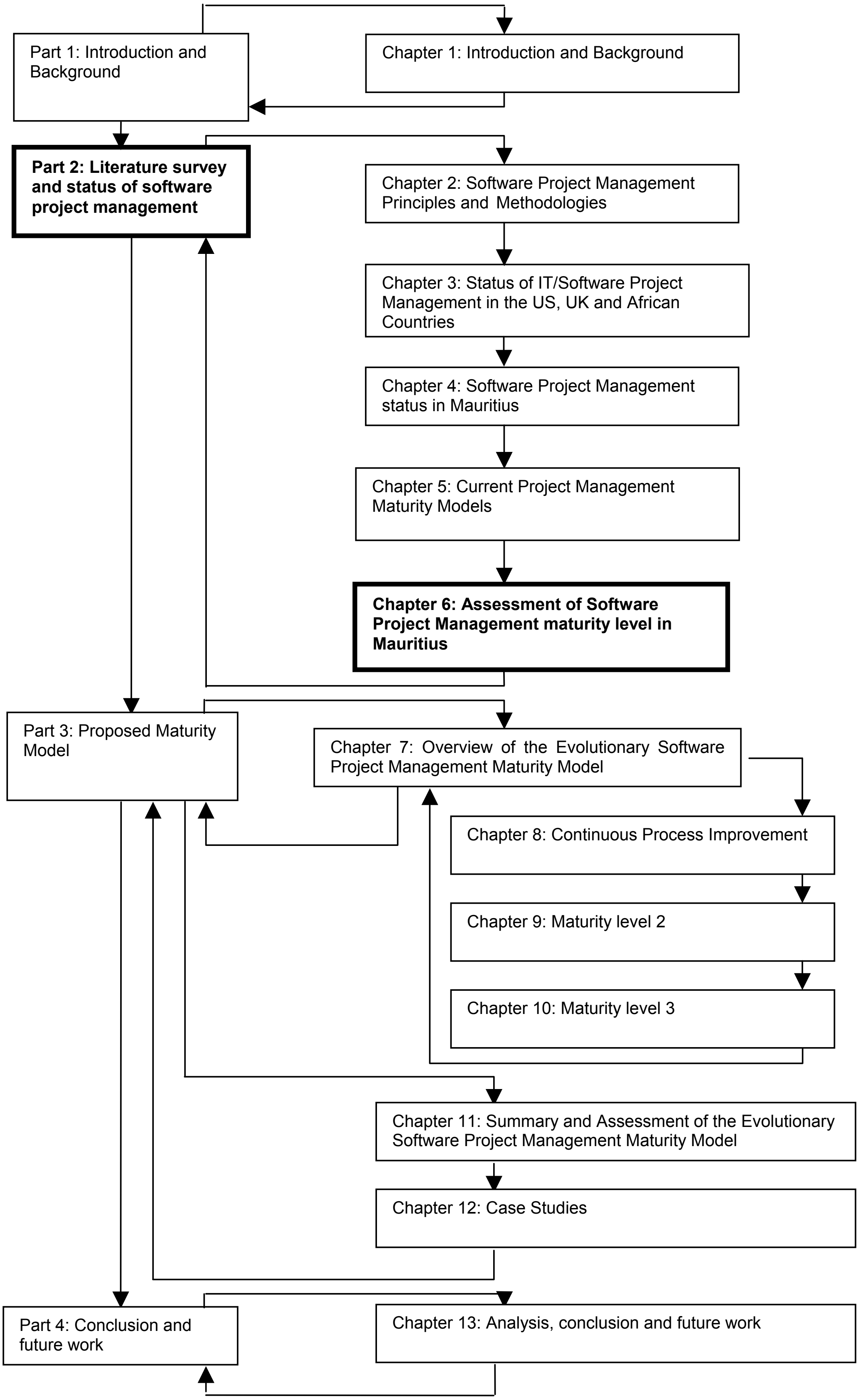


Figure 6.1: Chapter 6 in context within the overall thesis

6.1 Introduction

A survey conducted in Mauritius (Sukhoo et al., 2004) (elaborated in chapter 4) showed some of the shortcomings related to software project management. In order to improve the efficiency of software development, it is imperative for software companies in Mauritius to consider the management of projects in a streamlined manner.

This chapter takes the survey results one step further by assessing the *maturity* of software project management. Current project management maturity models in use were described in chapter 5 and the shortcomings of these models were highlighted in the Mauritian context. In Mauritius, few organisations in the software development sector were found to be even aware of a project management maturity model.

6.1.1 Aim, objectives and structure of this chapter

The aim of chapter 6 is to analyse the results of a survey conducted in Mauritius regarding an assessment of software project management maturity in Mauritius. The exercise was conducted in 2004 and the maturity level of software development organisations in Mauritius was determined. Such maturity level indicates the status of software project management in the country and also shows what is achievable in the future.

The deliverables, in terms of objectives, are the status of software development organisation in terms of the maturity level and improvement prospect.

The remainder of the chapter is structured as follows:

Section 6.2 – In this section, the results of the survey conducted by Sukhoo et al. (2005a) on software project management maturity in Mauritius are analysed. The results show that there is much room for improvement in order for the country to develop the IT sector as a pillar of the economy.

Section 6.3 – The conclusion of this chapter is described in this section. Respondents indicated an inclination towards the development of an indigenous software project management model for Mauritius that takes into consideration the needs of the country.

6.2 Assessment of software project management maturity in Mauritius

A maturity model can allow organisations to progress from one maturity level to the next at their own pace, depending on the availability of resources. Organisations adopting maturity models have been very successful on the international front. For instance, the US Department of Defence considers only software organisations that have reached maturity level 3 of the CMMI. Although it requires much effort and determination to reach maturity level 5 of the CMMI, at least level 3 would be a worthwhile initiative.

Basically, project management maturity and software project management maturity in particular is the progressive development of an enterprise-wide project management approach, methodology, strategy, and decision-making process (Project Management Experts, 2004). A number of maturity models were developed to allow organisations to progress at their own pace. A summary of maturity models is found in appendix C.

An assessment of software project management maturity can provide useful information on the level of success or failure of the sector. In a survey carried out in some Swiss organisations, Schiltz (2003) showed that project management maturity was associated with project success. Such a correlation was also demonstrated by Sonnekus and Labuschagne (2004). Therefore, in an attempt to achieve a higher rate of successfully completed projects, it is worthwhile to consider an improvement of the maturity level of an organisation.

Although not all companies use project management maturity models, it is possible to assess their maturity levels. One such framework was put forward by Kwak and Ibbs (2002). An assessment exercise for the maturity level of IT companies was carried out by Sonnekus and Labuschagne (2004) in the South

African context by the distribution of a questionnaire to such companies. These assessment methodologies provide useful information on the state of project management, particularly software project management, within a country.

A similar exercise as the one conducted by Sonnekus and Labuschagne (2004) in South Africa was carried out in Mauritius (Sukhoo et al., 2005a) to determine the country's software project management maturity level.

6.2.1 Aim of survey

The aim of the survey was to assess the software project maturity level of software development companies in Mauritius. In addition to the survey conducted to assess the status of software project management (described in chapter 4), this survey provides information that may be correlated to the level of success or failure in the sector.

6.2.2 Research methodology

Data collection was carried out during 2004 by issuing questionnaires to companies involved in software development in Mauritius. The companies were selected as per list of companies registered in Mauritius and published on the Website of the National Computer Board (Website: http://ncb.intnet.mu). A response rate of 29% was attained. One of the main reasons identified for the low response rate was that representatives from the companies were not interested in providing information. This problem was also faced during the previous survey. An attempt was made to interview the persons concerned but it failed because these persons could not find the time to contribute to this research.

The first section of the questionnaire of Sonnekus and Labuschagne (2004) was adopted as is. Not all questions in the second section were considered. A third section was added to gather information related to the social, economic, cultural and political aspects specific to Mauritius. The modified questionnaire is included in the section of published papers in this thesis.

6.2.3 Analysis of Survey Data

The following nine knowledge areas as per the PMBOK® standard (figure 2.3) were considered:

- Integration management.
- Scope management.
- Time management.
- Cost management.
- Quality management.
- Human resource management.
- Communications management.
- Risk management.
- Procurement management.

The above knowledge areas were mapped onto five maturity levels as per the information provided in the questionnaire prepared by Sonnekus and Labuschagne (2004) as follows:

Maturity Level 1 - Initial Process

- **Processes** - No established practices and standards.
- **Documentation** - Loose and ad-hoc.
- **Management** - Management understands the definition of a project, and is aware of the need for project management.
- **Metrics** - Collected informally on an ad-hoc basis.

Maturity Level 2 - Structured Process and Standards

- **Processes** - Processes exist, but are not considered an organisational standard.
- **Documentation** - Documentation exists on the basic processes.

- **Management** - Management supports the implementation of project management, but understanding and involvement is not consistent / applied to all projects. Large projects are executed in a systematic fashion, and management is involved in such projects.
- **Metrics** - Basic metrics to track cost, schedule and technical performance exist.

Maturity Level 3 - Organisational Standards and Institutionalised Process

- **Processes** - All project management processes are in place and established as organisational standards. These processes involve the clients as members of the project team. Nearly all projects use these processes.
- **Documentation** - Documentation exists on all the processes.
- **Management** - Management is regularly involved in input and approval of key decisions.
- **Metrics** - Metrics are formally collected and each project is evaluated and managed in light of other projects.

Maturity Level 4 - Managed Process

- **Processes** - Project management processes, standards and supporting systems are integrated with other corporate processes and systems.
- **Documentation** - Processes and standards are documented to support using metrics to make project decisions.
- **Management** - Management understands its role in the project management process. There are different management styles and project management requirements for different projects.
- **Metrics** - Efficiency and effectiveness metrics are used. All projects, changes and issues are evaluated based upon metrics from cost estimates, baseline estimates, and earned value calculations.

Maturity Level 5 - Optimizing Process

- **Processes** - Processes are in place and actively used to improve project management activities.

- **Documentation** - Lessons learned are regularly examined and used to improve project management processes, standards and documentation.
- **Management** - Management is focused not only on effectively managing projects but also on continuous improvement.
- **Metrics** - The metrics collected during project execution are used to understand the performance of a project and to assist in the making of organisational management decisions for the future.

The perceived and observed maturity levels were computed as follows:

Perceived Maturity Level versus Observed Maturity Level

The mean values of data from the various companies were used to interpret the perceived and observed maturity levels as follows:

$$M_{perceived} = \frac{\sum_{i=1}^{n} L_i}{n}$$

$$M_{observed} = \frac{\sum_{i=1}^{p} \left[\frac{\sum_{j=1}^{q} S_j}{q} \right]}{p}$$

where,

L = Perceived maturity level.

i = i^{th} company/individual.

L_i = Perceived maturity level of the i^{th} company/individual.

n = maximum number of responses received.

S = process maturity level.

j = j^{th} process.

Sj = Process maturity level of the j^{th} process.

p = maximum number of processes considered.

q = maximum number of processes with maturity levels between 1 and 5 (i.e. excluding responses which are not applicable).

The average maturity for an organisation was calculated by averaging the values of the nine knowledge areas. Then, the average for all organisations was calculated to arrive at the observed maturity. The results are shown in table 6.1.

Table 6.1: Average maturity level

Perceived maturity level	2.26
Observed maturity level	2.29
Difference	0.03

Respondents were requested to judge the overall maturity of their organisations based on the five maturity levels as defined above. An average value was computed to obtain the maturity level of Mauritian software development organisations. It was observed that there was a negligible underestimation by companies/individuals in the assessment of maturity levels. This difference of 0.03 is as shown in table 6.1.

The computed average maturity level of Mauritian software development companies was found to be 2.29 and this possibly indicated an effort to achieve maturity level 3. However, companies concerned need to pursue their efforts further so as to achieve a world-class standard, since these organisations were effectively operating at level 2. According to the description of maturity levels provided above, the processes, documentation, management and metrics are considered to be at the basic level. It is imperative for Mauritian software organisations to move to the next maturity level (i.e. level 3) in order to achieve further success in software project management.

Observed Maturity Level by Knowledge Area

The maturity level of each process area was computed and the results are as per table 6.2.

Table 6.2: Maturity levels by knowledge area

Process Area		**Average maturity level**
Integration Management		2.28
Core Functions	Scope Management	2.46
	Time Management	2.34
	Cost Management	2.50
	Quality Management	2.35
Average for Core Functions		***2.39***
Facilitating Functions	Human Resource Management	2.29
	Communications Management	2.25
	Risk Management	1.84
	Procurement Management	2.39
Average for Facilitating Functions		***2.19***

The histogram below shows the average maturity level for integration management, the core functions and the facilitating functions.

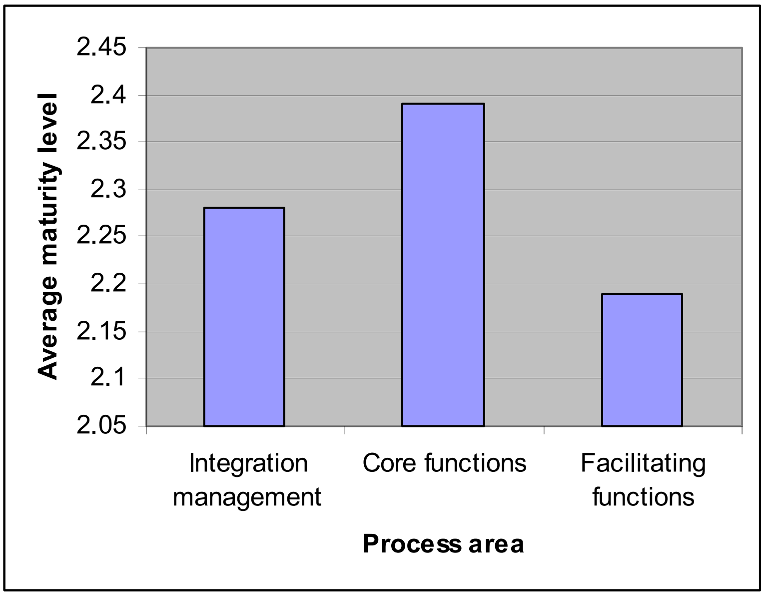


Figure 6.2: Average Maturity level by Process Area

The graph below shows more details by expanding the core and facilitating functions.

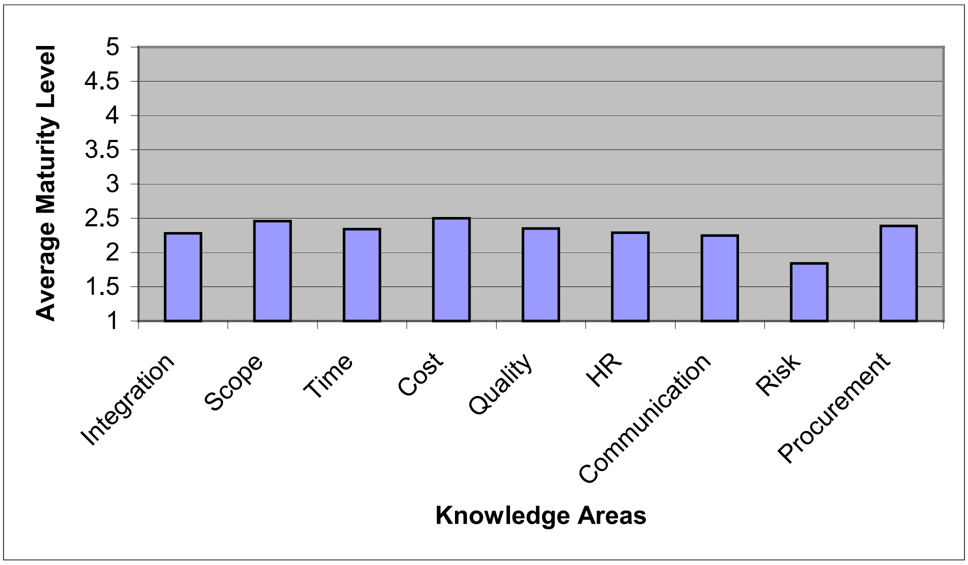


Figure 6.3: Average Maturity level by Knowledge Area

Most of the knowledge areas scored an average maturity level exceeding the value 2, except for Risk management. Risk management is important so as to prevent unanticipated problems during later stages of software development. It follows that Mauritian software development companies need to pay more attention to risk management as prevention is better than cure. Fixing problems during late stages can be very costly or even too late thus resulting in software project failure.

Cost management achieved the highest maturity level as compared to the other knowledge areas. Mauritian software development companies were, therefore, quite concerned about cost overruns. Therefore, metrics to track cost of projects were given high importance.

Scope management was equally treated with high priority given that this knowledge area had to be considered at an early stage. An incorrectly formulated project scope would certainly have major impacts on subsequent phases of the project. Therefore, it was necessary to carefully determine the appropriate scope so as to minimise the risk of the project getting off track during later stages.

Maturity Level by Process Groups

Each knowledge area of the PMBOK®, subdivided into a number of processes, was mapped onto the following five process groups:

- Initiating.
- Planning.
- Executing.
- Controlling.
- Closing.

The data so collected were further analysed and the results are summarised in table 6.3.

Table 6.3: Average maturity level by process group

Knowledge areas and processes	Initiating	Planning	Executing	Controlling	Closing
Integration Management					
Plan Development		2.33			
Plan Execution			2.33		
Integrated Change Control				2.17	
Scope Management					
Initiation	2.67				
Scope Planning		2.39			
Scope Definition		2.44			
Scope Verification				2.56	
Scope Change Control				2.22	
Time Management					
Activity Definition		2.32			
Activity Sequencing		2.35			
Activity Duration Estimating		2.47			
Schedule Development		2.44			
Schedule Control				2.12	
Cost Management					
Resource Planning		2.65			
Cost Estimating		2.59			
Cost Budgeting		2.53			
Cost Control				2.25	
Quality Management					
Quality Planning		2.25			
Quality Assurance			2.4		
Quality Control				2.4	
Human Resource Management					
Organisational Planning		2.11			
Staff Acquisition		2.44			
Team Development			2.31		
Communications Management					
Communications Planning		2.33			
Information Distribution			2.17		
Performance Reporting				2.21	
Administrative Closure					2.29
Risk Management					
Risk Management Planning		1.92			
Risk Identification		2.00			
Qualitative Risk Analysis		1.69			
Quantitative Risk Analysis		1.69			
Risk Response Planning		1.92			
Risk Monitoring and Control				1.83	
Procurement Management					
Procurement Planning		2.33			
Solicitation Planning		2.28			
Solicitation			2.44		
Source Selection			2.33		
Contract Administration				2.47	
Contract Closeout					2.5
Average for each process group	**2.67**	**2.26**	**2.33**	**2.25**	**2.40**

Each process group attained an average maturity level between 2 and 3 (last row of table 6.3), thereby showing some consistencies with previously computed averages. The average maturity level for each process group is shown graphically in figure 6.4.

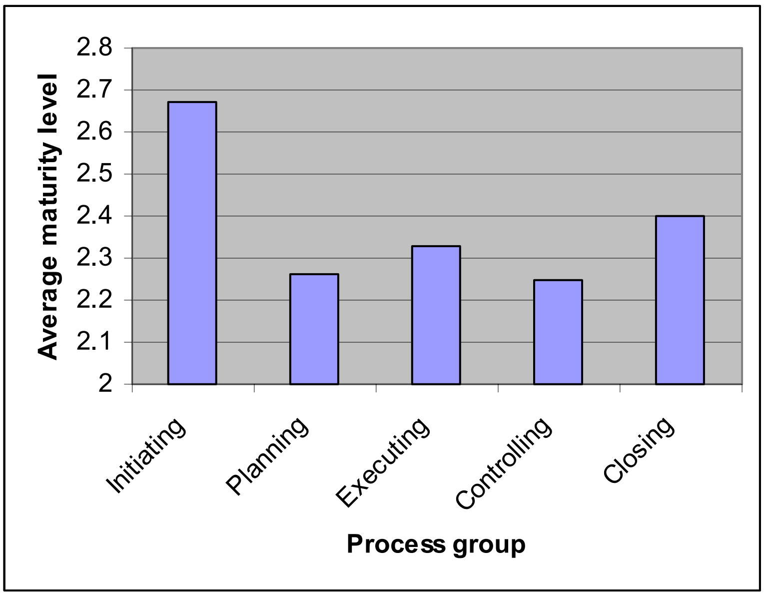


Figure 6.4: Average Maturity level by process group

It was observed that the initiating process group was carried out well as compared to the other process groups. This could be explained by the amount of work involved and the fact that software developers are in a hurry to get the job done.

Maturity level to cope with Factors such as Culture, Politics and Economy

The perceived maturity level versus observed maturity level to cope with other factors like culture, politics, climate and the economy were computed and summarised in table 6.4.

Table 6.4: Average maturity level

Perceived maturity level	1.95
Observed maturity level	2.12
Difference	0.17

These factors were considered since the research conducted by Muriithi and Crawford (2003) showed that project management in developing countries was different from that in Western countries. Therefore, the initiative was taken to assess the maturity of Mauritian organisations to deal with such factors.

According to the survey result, the maturity to deal with cultural, political and economic factors was at level 2 (as per the observed maturity level in table 6.4). Progress to the next level would certainly be desirable.

6.3 Conclusion

In this chapter, the results of a survey on software project management maturity were analysed. Although not all organisations used a maturity model, a framework based on a combination of the PMBOK® and the CMM was adopted to evaluate the maturity level of Mauritian software development organisations. It was found that the maturity of Mauritian software organisations attained a level of 2 on a scale of 1 to 5. Therefore, Mauritian organisations are still at the basic level and efforts necessary to reach a higher level are obvious.

It was also demonstrated from other researches that project success or failure was linked to the maturity level of an organisation (Schiltz, 2003). The higher the maturity level, the more successful were the organisations.

In order to assist Mauritian organisations to achieve higher maturity levels, a software project management maturity model is proposed in the next chapter. Although there are several maturity models, it was observed during the surveys that respondents were in favour of an indigenous model that could take into consideration the requirements of Mauritian software organisations.

In chapter 7, an evolutionary approach to software project management is proposed for Mauritius. An overview of such an evolutionary software project management model is provided and the details are discussed in the chapters that follow.

Part 3

Proposed maturity model

Chapter 7

Overview of Evolutionary Software Project Management Maturity Model

Table of contents

7.0 Chapter layout

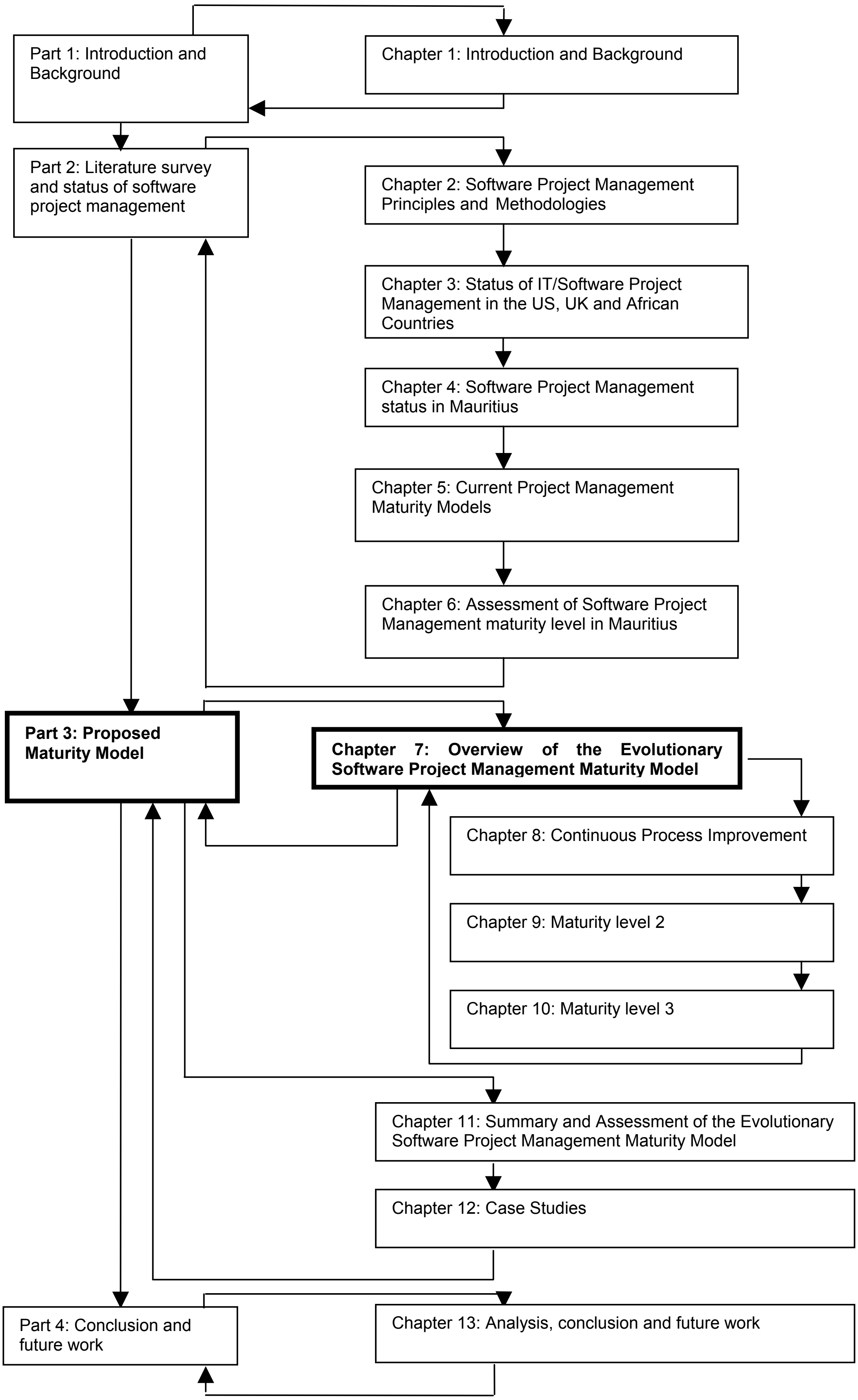


Figure 7.1: Chapter 7 in context within the overall thesis

7.1 Introduction

In part 2 of this thesis, the scene was set for the importance of a software project management approach of an indigenous nature for developing countries and Mauritius in particular. In order to complement the studies conducted in developing countries, surveys were conducted in Mauritius to gather information about the status of software project management. The political, economic, social and cultural problems were found to be different from developed countries. In addition, Mauritius has certain specificities, namely a strong political commitment towards Information and Communication Technologies. It is a small island located relatively far away from other countries. It has a shortage of skilled staff and relies on unstable economic pillars like sugar industry, textile products and tourism.

In chapter 6 an assessment of software project management maturity in Mauritius was presented. Even results from software project management approaches not based on maturity models were mapped onto such models. Given that maturity levels were found to be linked to the level of project success, it can be deduced that a maturity model-based software project management approach can directly yield information on the success of an organisation.

In this chapter an introduction is provided on a proposed software project management model to address problems encountered in Mauritius. This model, called "Evolutionary Software Project Management Maturity Model", ($ESPM^3$), may assist software development organisations by adopting a progressive implementation of various processes to the management of software projects.

$ESPM^3$ is introduced broadly by starting with research that led to its development. The different maturity levels are also briefly described together with respective process areas that are necessary. Maturity levels are also given a mathematical representation.

In section 7.3 three popular methodologies/models/standards are analysed critically regarding their unsuitability for software project management in Mauritius. PMBOK®, PRINCE 2® and CMMI have been specifically chosen for this analysis given that they were found to be known to software developers in Mauritius according to the survey conducted by Sukhoo et al. (2004b). In addition, CMMI is also known to have been used by one organisation in Mauritius.

7.1.1 Aim, objectives and structure of this chapter

The aim of chapter 7 is to provide an overview of the $ESPM^3$ before the details are discussed in subsequent chapters. The motivation and results that helped in the development are also explained.

The remainder of the chapter is structured as follows:

Section 7.2 – In this section, the motivation for the development of a software project management model is explained. Studies carried out by various researchers are presented. These studies demonstrated the need for a new model of an indigenous nature for the management of software projects in African developing countries. Problems related to project management in these countries are also mentioned.

Section 7.3 – The unsuitability of methodologies like PRINCE 2®, PMBOK® and CMMI for Mauritius is discussed in this section. The discussion is supported by research that we conducted in Mauritius as well as shortcomings that could be identified in these methodologies/models/standards.

Section 7.4 – In this section, the survey conducted by Sukhoo et al. (2007) is presented. Key process areas are identified and classified into various maturity levels.

Section 7.5 – In section 7.5, the various maturity levels in ESPM^3 are briefly described. Diagrammatic representations are used to clarify the concepts and mathematical representations are also considered.

Section 7.6 – The suitability of ESPM^3 is explained in section 7.6 while taking into account the shortcomings encountered with other methodologies/models/standards.

Section 7.7 – On the basis of problems relevant to developing countries, the suitability of the ESPM^3 is indicated by taking into consideration the solution provided. Soft skills and the environmental KPAs are speculated to be of interest for developing countries.

7.2 Motivation for a new software project management methodology/model/standard

There is ample evidence that the inherent principles of project management have been practiced already for thousands of years (Kwak, 2003). The results are that many human achievements, for example the pyramids of Egypt and the Great Wall of China, are recognised as wonders of the present-day world. It is also unfortunate that the Manhattan project, although considered as a successful human undertaking (the development of the atomic bomb), at a later stage caused the loss of lives on a large scale (Cleland, 2004).

Apart from the construction and engineering areas, applied principles of project management made inroads into virtually all avenues of work by encompassing sectors such as the military, social and community development, medicine, agriculture and education to name just a few (Kwak, 2003). Today, the impact of project management is most profound in the area of Information Technology (Gray and Larson, 2000) where new hardware and software products are constantly flooding the world market. Increased pressure to reduce cost and delivery time in a highly global and competitive environment has given due credit to project management principles, techniques and tools. Software project management, in particular, is an area of research with a view to achieve higher levels of quality and to improve on both cost and schedule estimates. This is evident from frequent new releases of project management methodologies/models/standards like PMBOK® (PMBOK® Guide, 2006), PRojects IN Controlled Environments (PRINCE) (CCTA, 2002) and the Capability Maturity Model Integration (CMMI Product Team, 2002). Project management methodologies/models/standards were studied prior to the development of a suitable methodology/model/standard for Mauritius.

A software project management model is being proposed since it has been noted that a software project has certain specificities that are different from traditional engineering projects. Hughes and Cotterell (2006) argued in favour of the specificities of managing software projects. Some pertinent differences between software projects and traditional engineering projects were also elaborated on by Sukhoo et al. (2004a).

A plethora of project management methodologies/models/standards is now available and many of them are subject to continuous improvement. Project management methodologies/models/standards are generally generic in nature, in that they cut across various disciplines and are used in many countries. Some studies, for example by Muriithi and Crawford (2003) as well as Stuckenbruck and Zomorrodian (1987), have revealed that such methodologies/models/standards are not necessarily universally applicable because factors like economic rationalities are too often assumed and legal, political, cultural and religious variations occur across nations and cultures.

Hofstede's dimensions as applied to African countries (Muriithi and Crawford, 2003), Kuwait (Aladwani, 2002) as well as Mexico, India and Russia (Rao, 2004) have been discussed in terms of high power distance and high uncertainty avoidance as compared to Western/European countries. Open discussions are not favoured and low tolerance for ambiguities are noted in developing countries. It is plausible that these factors may negatively influence the successful outcome of software projects, particularly in developing economies, as these factors are seldom taken into account by a particular methodology/model/standard.

Developing countries constantly face problems of insufficient skilled manpower, funds, and political and social incentives (Muriithi and Crawford, 2003). Mauritius, for example, is facing such problems at a crucial moment in its history as the Government has embarked on a vision to develop the island into a "Cyber Island" while at the same time allowing Information and Communication Technologies (ICT) to emerge as the fifth pillar of the economy besides sugar, tourism, textile and the financial services sector (Eid, 2002). In particular, some of the problems affecting the Mauritian IT sector are:

- Lack of skilled human resources (BPO Secretariat, 2005).
- High labour mobility (BPO Secretariat, 2005).
- Lack of training (BPO Secretariat, 2005).
- Ageing population.
- Unstable economic situation (Muriithi and Crawford, 2003).
- Project management approaches of Western/European origins are not completely suitable and are too expensive to adopt for most organisations (Muriithi and Crawford, 2003).
- Lack of management commitment.
- Lack of soft skills in management of projects.
- Organisations are characterised by tall hierarchical structures (Muriithi and Crawford, 2003).
- Existing methodologies/models/standards are considered to be too bureaucratic.

The above issues will now be motivated by looking at the major methodologies/models/standards.

7.3 Unsuitability of PMBOK®, PRINCE 2® and CMMI for Mauritius

According to the survey of project management tools, techniques and methodologies in Mauritius (Sukhoo et al., 2004b), it was noted that 50% of respondents were aware of PRINCE 2® and only 20% were aware of PMBOK®. Most organisations did not use any formal methodologies/models/standards for several reasons, among which are the high cost of implementation, bureaucratic nature and unsuitability of such methodologies/models/standards.

PRINCE 2® does not consider human resources management and procurement management. These process areas (or knowledge areas) are expected to be treated separately as they are thought not to be part of a project management methodology. Socio-economic and environmental influences have been left out by PRINCE 2®. No organisations in Mauritius have been identified to be using PRINCE 2® fully.

Although in the PMBOK® Guide, it is spelt out that a project team should consider a project in the context of political, economic, demographic and cultural norms, it does not explain how or when to handle these. The PMBOK® Guide, furthermore, revealed an increase in the number of pages from around 200 to around 400 since the previous release. The number of processes involved also increased from 39 to 44, thereby adopting a rather complex systems view of project management in general, and software project management in particular (Wideman, 2005). Specifically, small projects can be affected by the number of processes involved. The project team may, therefore, be overloaded by the number of process areas. In Mauritius, the lack of skilled software developers within organisations coupled with high labour mobility has a marked influence on the applicability of PMBOK®. The survey conducted by Sukhoo et al. (2004b) did not reveal the use of PMBOK® by any organisation in Mauritius.

CMMI provides progressive improvements of organisational maturity by achieving maturity at a pre-defined level before advancing to a higher level.

Hence, the project team focuses on a restricted set of process areas to reach a maturity level. However, CMMI does not assist in the management of influences brought about by social, economic or environmental pressures as well as human resource problems. In addition, the model incorporates a sheer volume of documentation of over 600 pages that increases the learning duration. Given that human interaction is important for a successful software development process, CMMI alone cannot address the human resource management problem. In Mauritius, not many companies can afford to adopt CMMI given the high initial cost of training required. Thus far, only one Mauritian organisation (The State Informatics Limited) is using this model (SIL, 2006).

According to the survey on project management tools, techniques and methodologies used in Mauritius (Sukhoo et al., 2004b), it was observed that there was resistance to adopt approaches of Western/European origin. This is also in line with the research conducted by Muriithi and Crawford (2003).

None of the above methodologies/models/standards treated the political, cultural, social and economic aspects fully. Muriithi and Crawford (2003) as well as the survey conducted by Sukhoo et al. (2004b) revealed the need for indigenous project management methodologies/models/standards in African developing countries. This initiative is being considered specifically for software in this thesis and to this end the proposed Evolutionary Software Project Management Maturity Model (ESPM^3) takes into consideration the specific requirements for developing countries and Mauritius in particular.

7.4 Survey/interview conducted in Mauritius

In addition to the two surveys discussed in chapter 4 and 6, a combined survey/interview was conducted with 11 pertinent software development organisations in Mauritius during the period July 2006 to August 2006 so as to collect data on key process areas (KPAs), or focus areas, deemed necessary for managing of software projects. All the respondents agreed on the applicability of a maturity model with less than 5 maturity levels for Mauritius.

Although it can be argued that a maturity model may hide certain deficiencies, an organisation is free to consider all KPAs for the proposed maturity model. The different levels help an organisation to concentrate on certain KPAs at a level until it attains maturity. Then, the next level is considered. Organisations are able to identify their strengths and weaknesses and consider their areas of improvement. In the case of CMMI, which has 5 maturity levels, not many organisations worldwide have reached the highest level. Therefore, the proposed model is meant to allow all Mauritian software development organisations to attain the highest level in the shortest time possible in order to develop the Information and Communication Technologies sector. Even PRINCE 2 maturity model has adopted less than 5 levels, namely 3.

Three maturity levels (levels 1 to 3) together with a set of continuous improvement key process areas were identified to be an appropriate number of maturity levels. Key process areas (KPAs), i.e. focus areas required to achieve a particular maturity level, were also identified and eventually classified into different maturity levels.

A list of KPAs was provided in the questionnaire and most respondents agreed to its applicability. In this context, the most applicable KPAs identified in the Mauritian context were found to be the following:

- Time management.
- Cost management.
- Quality management.
- Human Resource management.
- Risk management.
- Soft skills management.
- Contract management.
- Change management.
- Software specific focus.
- Integration management.
- Environmental management.

Respondents classified the KPAs into two levels and a continuous process improvement group as per table 7.1:

Table 7.1: Survey/Interview results for classification of KPAs

	Respondents classification of KPA's at the different levels			Total no of respondents	Percentage response
	Level 2	Level 3	Continuous process improvement		
Time management	8	0	3	11	100%
Cost management	5	2	3	10	91%
Quality management	7	1	3	11	100%
Human resource management	1	7	2	10	91%
Risk management	2	7	1	10	91%
Soft skills management	2	3	5	10	91%
Contract management	2	7	0	9	82%
Change management	1	3	7	11	100%
Software specific focus	3	1	6	10	91%
Integration management	0	1	10	11	100%
Environmental management	9 out of 11 mentioned that a KPA dealing with cultural, political and economic aspects was important.				82%

It was noted that some respondents did not provide any classification for some KPAs. It can, however, be noted that above 80% of responses were received for the classification of the KPAs. For Time management, 8 respondents mentioned that the KPA had to be considered at level 2, no respondent mentioned that the KPA had to be at level 3 and 3 respondents mentioned that the KPA had to be classified into the continuous process improvement group. Therefore, Time management was considered as a maturity level 2 KPA (that is, the level at which the majority of respondents agreed that the KPA should be). The continuous process improvement group, however, comprises KPAs that need to be taken into consideration at all maturity levels (at least informally at level 1 as well).

According to the combined survey and interview carried out by Sukhoo et al. (2007), and taking cognizance of the preference of the people in developing countries, for instance Mauritius, three maturity levels were fixed and the key process areas identified were categorised into two levels (2 and 3) as mentioned previously. Other KPAs were found to be important at all levels and they act as factors that initiated improvements for KPAs at both levels 2 and 3.

For instance, Change management can have an impact on the Time management KPA in case political pressure is exerted to motivate project delivery before the scheduled time.

The different KPAs are, therefore, grouped into maturity levels as shown in table 7.2.

Table 7.2: KPAs grouped in maturity levels

KPA	**Maturity level**
Time management	2
Cost management	2
Quality management	2
Human resource management	3
Risk management	3
Soft skills management	Continuous process improvement group
Contract management	3
Change management	Continuous process improvement group
Software specific focus	Continuous process improvement group
Integration management	Continuous process improvement group
Environmental management	Continuous process improvement group

Most of the above KPAs were also considered to be important success factors in a previous study conducted in Mauritius (Sukhoo et al., 2005a). Furthermore, each KPA is mapped onto four process groups as per ISO 9001:2000 (Mauritian standard, 2001) requirements as follows:

(i) *Plan*: Establish the objectives and processes necessary to deliver results in accordance with customer requirements and organisational policies.

(ii) *Do*: Implement the processes.

(iii) *Check*: Monitor and measure processes and products against policies, objectives and requirements and report the results.

(iv) *Act*: Take actions to continually improve process performance.

ISO 9001:2000 was adopted as the quality management standard for our proposed ESPM[3] because many organisations in Mauritius are already following this standard (Management Audit Bureau, 2005). Following the identification and grouping of the KPAs into different maturity levels, a conical structure is used to represent the maturity levels comprising these KPAs.

7.5 The Evolutionary Software Project Management Maturity Model (ESPM[3])

The conceptual representation of the proposed ESPM[3] adopts a conical structure with three defined levels (maturity levels 1 to 3), depicted in figure 7.2.

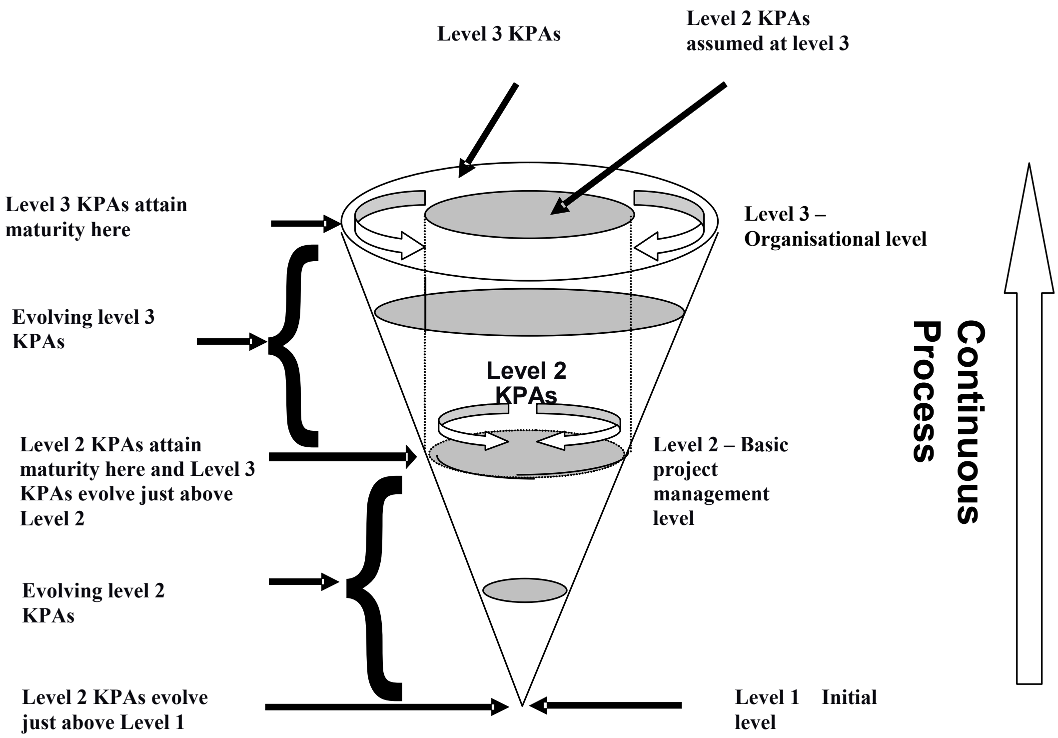


Figure 7.2: Conceptual representation of ESPM[3]

Maturity levels 2 and 3 each consist of a number of key process areas (KPAs), which are focus areas aimed at achieving the respective maturity level. Maturity level 1 is the initial level of the proposed model and has no KPAs associated with it since software development at this level is carried out in an ad-hoc manner. In other models (e.g. in the Berkeley Project Management Process Maturity Model and the Microframe's self assessment tool), this level is often considered as the “chaotic” or “ad-hoc” level. The KPAs at levels 2 and 3 are represented conceptually as sectors on the horizontal section of the cone (figure 7.3).

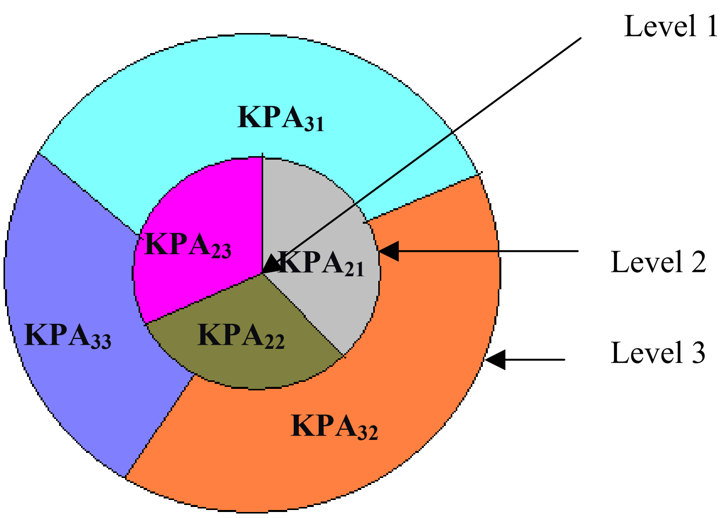


Figure 7.3: KPAs at maturity levels 2 and 3 (viewed from above)

The above representation is discussed in more detail in the following sections. KPAs at a specified maturity level are assumed at higher levels. For example, at maturity level 3, a cross section of the cone reveals the KPAs at level 2 as well as the new KPAs at level 3. In figure 7.3, at maturity level 3, the relevant KPAs are KPA_{31}, KPA_{32} and KPA_{33} as well as the KPAs of maturity level 2 which are KPA_{21}, KPA_{22} and KPA_{23}. It is to be noted, however, that the pictorial representation of an area of a sector is not representative of the amount of effort required to implement a specific KPA.

Specific to problems identified in some Mauritian software development companies, a range of key process areas was identified to be implemented at the various maturity levels introduced above. Specific shortcomings pertaining to some or all of these KPAs were identified during surveys conducted in Mauritius on project management tools, techniques and methodologies (Sukhoo et al., 2004b) and an assessment of software project management maturity (Sukhoo et al., 2005a). Time management, Cost management and Quality management required much improvement. Risk management, inappropriate project team, poor planning, lack of skilled professionals, mobility of labour, communication problems among team members as well as with clients, and the lack of an appropriate methodology were among several problems that were encountered. Furthermore, Risk management was not adequately addressed, since most companies surveyed at that point in time were found to be at maturity level 1 as project management was being conducted in an ad-hoc manner.

7.5.1 Representation of KPAs throughout the Evolutionary Process

There are some KPAs that are not tied to any specific maturity level but need to be considered throughout the evolutionary process. This continuous process improvement group of processes spans over all three levels (i.e. maturity levels 1, 2 and 3). The *continuous process improvement* group, M_c, is defined generally as the union of a number of KPAs, i.e.

$$M_c = \bigcup_{k=1}^{P} KPA_{ck} \qquad (7.1)$$

where p is the total number of KPAs in the continuous process improvement group. A KPA in M_c is identified by an additional subscript "k" to the specific KPA, e.g. KPA_{c1} is the first KPA in M_c, KPA_{c2} is the second KPA in M_c and so on.

For $ESPM^3$, 5 KPAs are identified that need to be considered at all maturity levels. This selection was based on the respondents' classifications of the KPAs as per the questionnaire that was distributed and in combination with interviews that were conducted.

The KPAs identified for the continuous process improvement group are:

- Soft skills management (KPA_{c1}).
- Change management (KPA_{c2}).
- Software specific focus (KPA_{c3}).

- Integration management (KPA_{c4}).
- Environmental management (KPA_{c5}).

The above KPAs are expected to influence all other KPAs and induce continuous improvement. They are represented by the arrow on the right in figure 7.4.

The Environmental management KPA is concerned with specific social, cultural, political and economic factors for Mauritius. These factors were found to affect software project management. For example, a project deemed to offer certain political benefits needs to be given high priority and must be managed with extreme care. In managing the environmental factors, the following inputs are transformed into outputs:

- List of environmental factors to be addressed (for example political, social, cultural and economic benefits so that such projects are given priority). An example of such a project is treated in the case study (see chapter 12).
- List of ways to deal with the factors (for example, assigning skilled human resources and technology).
- Historical data.

The transformation yields the following outputs:

- Increase the possibility of satisfied team members and stakeholders.
- Safe software development environment (for example, other projects must not fail due to political, social and economic priorities).
- Lessons learnt.

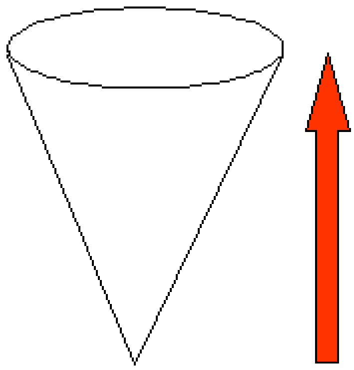


Figure 7.4: Continuous process improvement

Instantiating p in formula (7.1) for the number of KPAs at this level, the continuous process improvement group for ESPM[3] is given by:

$$M_c = \bigcup_{k=1}^{5} KPA_{ck} \qquad (7.2)$$

The continuous process improvement group for ESPM[3], M_c, is the union of Soft skills management (KPA_{c1}), Change management (KPA_{c2}), Software specific focus (KPA_{c3}), Integration management (KPA_{c4}) and Environmental management (KPA_{c5}).

7.5.2 Maturity Level 1

Maturity level 1 is the *initial level* in the proposed model as shown in figure 7.5.

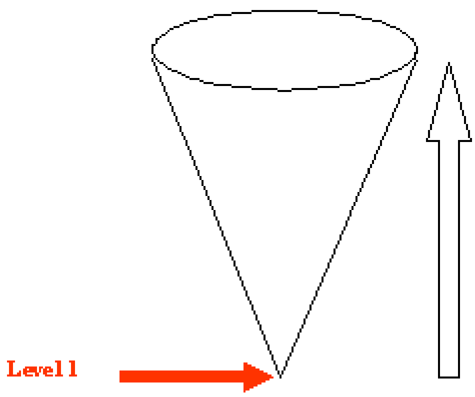


Figure 7.5: Maturity level 1

Similar to other existing maturity models, ESPM[3] includes maturity level 1 in order to accommodate organisations that do not have any formalised methodology/models/standards. At this level, no KPAs are defined. The software development process is carried out in an ad-hoc (possibly chaotic) manner similar to other maturity models and project success depends on the project team. A change in the project team while the project is in progress may disrupt its execution as the skills and processes used are not consistent among project teams. The operation of an organisation at level 1 involves uncalculated risks and progress towards level 2 is necessary. Maturity level 1, namely M_1, contains no KPAs, i.e.:

$$M_1 = \varnothing \tag{7.3}$$

Even the continuous process improvement group of KPAs are not yet developed (or not formalised) at this level.

7.5.3 Maturity Level 2

This level provides a basic project management focus, which is important for the software development process, since it forms the foundation for level 3. It is referred to as the basic project management level. Figure 7.6 depicts level 2 as a horizontal section of the cone.

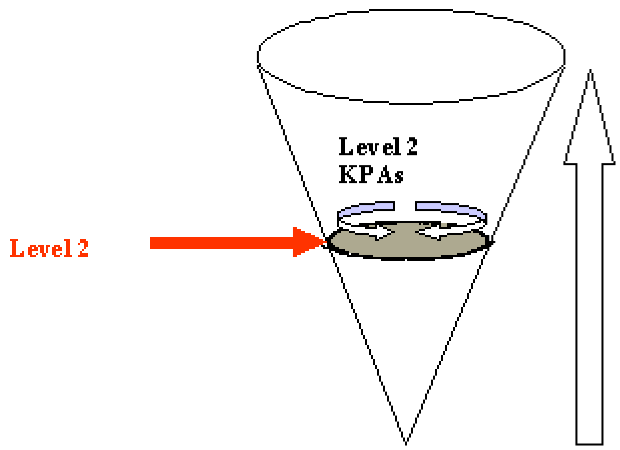


Figure 7.6: Maturity level 2

Maturity level 2, namely M_2, is the union of all KPAs at level 1 (M_1), all KPAs at the current level and the continuous improvement process group of KPAs (M_c), i.e.:

$$M_2 = M_1 \cup \bigcup_{j=1}^{n} KPA_{2j} \cup M_c$$

M_1 and M_c are given by (7.3) and (7.1) respectively and n is the total number of KPAs at level 2.

Figure 7.7 shows graphically how the KPAs (say n KPAs) at level 2 are represented as a view from the top of the cone. In addition, the continuous process improvement group of KPAs, M_c, influences the level 2 KPAs. For example, integration management also takes into consideration the level 2 KPAs. In this way, the continuous process improvement group KPAs as well as the level 2 KPAs operate together coherently instead of in isolation. Similarly, the software specific focus KPA affects time management by reducing the development time through the software reuse technique.

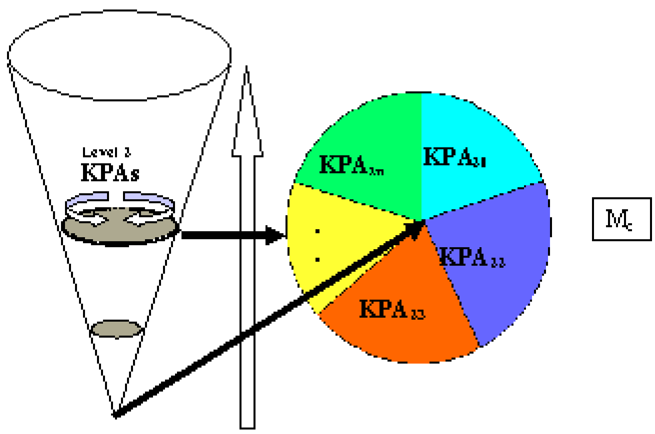


Figure 7.7: Conceptual representation of n KPAs at levels 1 and 2

Assuming a maturity level with n KPAs, each one is represented as a sector by viewing the appropriate horizontal section of the cone. At a specified maturity level, its associated KPAs are considered to have evolved fully and do not require further development at any of the higher levels except for continuous process improvement where improvement is required for the process to adapt to the dynamic environment in which organisations operate.

A KPA at level 2 is indicated as KPA_{2i}, e.g. KPA_{23} is the third KPA at maturity level 2. Only the essential KPAs for the basic project management level are selected based on the respondent classification in the survey conducted in order to encourage organisations to progress rapidly to level 2 and ensure that projects are delivered on time, within budget and according to a specified quality standard. In this context, for $ESPM^3$, based on respondents' classifications (see Table 7.2), three KPAs are chosen to achieve this level and they are:

- Time management: to ensure timely delivery of projects (KPA_{21}).
- Cost management: to deliver projects within budget (KPA_{22}).
- Quality management: to deliver projects according to ISO 9001:2000 standards (KPA_{23}) and to meet customer requirements.

Substituting for n in formula (7.3) and for $M_1 = \emptyset$, M_2 for $ESPM^3$ is defined as

$$M_2 = \bigcup_{j=1}^{3} KPA_{2j} \cup M_c \qquad (7.4)$$

Following the combined survey/interview that was conducted, one respondent mentioned that cost management was not relevant for his organisation as all software development was performed in-house. In such a case, this KPA may be ignored in the $ESPM^3$. However, the KPA may also be included in similar organisations by considering the salary of the software developers. The project manager may assess the productivity of the software developers.

7.5.4 Maturity Level 3

This level provides an organisational focus. Maturity level 3, M_3, is the union of all KPAs at level 3 and level 2 (M_2), where M_2 already includes the continuous improvement process group of KPAs (M_c). This level is shown in figure 7.8.

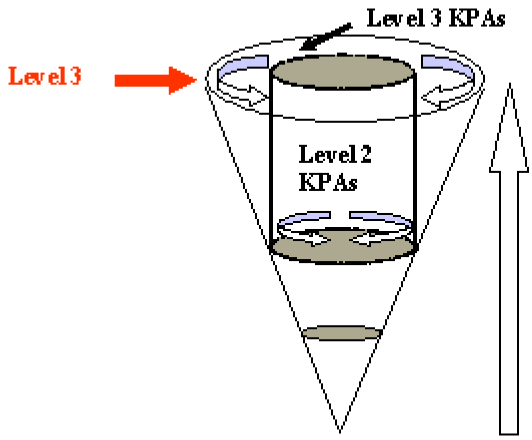


Figure 7.8: Maturity level 3

Therefore,

$$M_3 = \bigcup_{i=1}^{m} KPA_{3i} \cup M_2 \qquad (7.5)$$

where m is the total number of KPAs at level 3 (the KPA_3's) and M_2 is given by (7.4).

In general, for ESPM[3] containing any number of maturity levels, the i^{th} maturity level, i.e. M_i, is defined as:

$$M_i = \bigcup_{j=2}^{i-1} \{ M_j - M_c \} \cup \left\{ \bigcup_{K=1}^{M} KPA_{ik} \right\} \cup M_c \qquad (7.6)$$

where i, j, k and m are natural numbers, M_c is the continuous improvement process group of KPAs and KPA_{ik} is the k^{th} key process area associated within the i^{th} level.

KPAs for level 3 are normally developed once level 2 has been reached and the organisation concerned intends to reach level 3. In this model specific to the Mauritius context as an instance of developing countries and guided by respondents' classifications summarised in table 7.1, the following KPAs are developed at level 3:

- Human Resource management (KPA_{31}).
- Risk management (KPA_{32}).
- Contract management (KPA_{33}).

Instantiation of the KPAs at levels 1, 2 and 3 leads to the arrangement of the KPAs depicted graphically in figure 7.9.

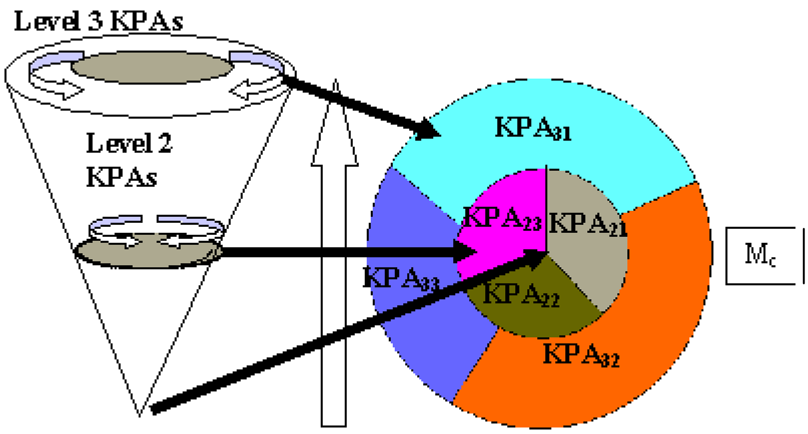


Figure 7.9: Instantiated representation of KPAs at levels 1,2 and 3

Substituting for m in formula (7.5) defines M_3 for $ESPM^3$ as:

$$M_3 = \bigcup_{i=1}^{3} KPA_{3i} \cup M_2 \qquad (7.7)$$

Here KPA_{21}, KPA_{22} and KPA_{23} are the key process areas that attained maturity at level 2 (M_2), while KPA_{31}, KPA_{32} and KPA_{33} are the key process areas that attained maturity at level 3. In general, therefore, KPAs at level i evolve after efforts have been made to reach level i from level i-1. Thus, any KPA_{in} for valid values of i and n, matures at level i.

An organisation is, however, not prevented from developing and using a KPA at a higher level. For example, an organisation at level 2 may include a KPA of level 3 in the case that this KPA is considered to be important. Note that the organisation cannot claim to be fully at the higher level if this happens, but it can be referred to as being at an extended level. For example, an organisation at level 2 that is using a KPA of level 3 will be referred to as level 2-extended.

7.5.5 Mapping of KPAs to the PDCA cycle

As mentioned before, each KPA is mapped to the PDCA (Plan-Do-Check-Act) cycle. This is shown in figure 7.10.

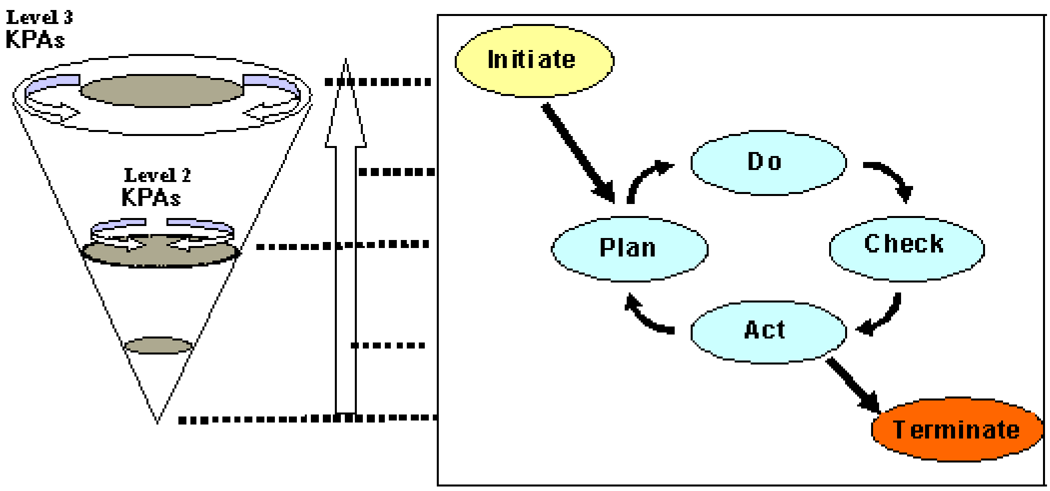


Figure 7.10: Mapping of KPAs to the PDCA cycle

The initiate process group marks the start of the cycle while the terminate process group marks the end of the cycle.

7.6 Suitability of ESPM^3 for Mauritius

The proposed ESPM^3 addresses the problem of human resources, soft skills, contract management as well as software specificities. It also takes into consideration the political, social, economic and cultural aspects associated with software project management. In the Mauritian context, it allows software development organisations to focus on a few key process areas initially (e.g. 3 KPAs till maturity level 2 is reached) and subsequently adding more KPAs to progress to the next higher level (i.e. level 3). An organisation at a lower level may also include a KPA of a higher level and enable it to operate at the extended lower level. The continuous process group comprises those KPAs that affect KPAs at levels 2 and 3. For example, Integration management binds all KPAs in a coherent manner and the Environmental management KPA makes use of the political factor, when relevant, to influence Time management.

The Software specific focus KPA influences the way project management is carried out for software development, e.g. prototyping is considered as an important aspect in software development. KPAs are mapped onto the Plan-Do-Check-Act process group as per the ISO 9001:2000 requirement (Mauritian standard, 2001). PMBOK® and CMMI both include numerous key process areas (in the case of PMBOK®, the KPAs called knowledge areas are further subdivided into 44 processes) to be considered and, therefore, the implementation of such methodologies/models/standards is time consuming and renders the software development process complex in the Mauritian context. Thus, the ESPM^3 is an attempt to reduce much of the bureaucracy and complexities associated with other models.

We are of the opinion that the evolutionary process is better shown in ESPM^3 (e.g. in figure 7.1) than in CMMI. The conical structure shows the growth of KPAs as in figure 7.11.

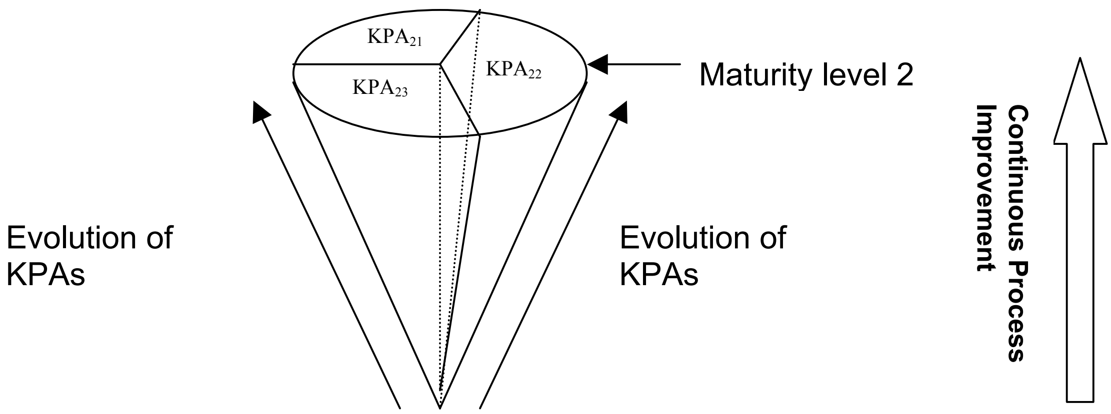


Figure 7.11: Evolutionary process of maturity level 2 KPAs

7.7 Conclusion

In this chapter, reference was made to some of the problems experienced by Mauritius as a developing country with respect to the delivery of successful projects. Mauritius is trying to develop its ICT sector as a pillar of its economy with the result that the number of software development projects is increasing. Software project management is, therefore, expected to play an important part in the development of software. Based on earlier research, it was shown that project management approaches of European/Western origins are not universally applicable (Muriithi and Crawford, 2003). In addition, previous work carried out by Sukhoo et al. (2004b) and Sukhoo et al. (2005a) illustrated that this was also the case in Mauritius.

The ESPM^3 provides a simple and flexible evolutionary approach for managing software development projects by portraying the evolutionary development of KPAs and at the same time rendering conceptually an integrated management process. The software specific focused KPA placed special emphasis on software projects by considering the strengths and weaknesses of such projects (Sukhoo et al., 2004a). The soft skills KPA was included to emphasise on the importance of human relationships in the management of software projects. In addition, KPAs were considered with particular attention to the economic, cultural, social and political aspects (Environmental KPA) to improve the success rate of software projects.

The next three chapters cover in more detail the continuous process improvement group of KPAs and maturity levels 2 and 3 along with their KPAs. The inputs, processes and outputs are also discussed.

Chapter 8

Continuous Process Improvement

Table of contents

8.0 Chapter layout

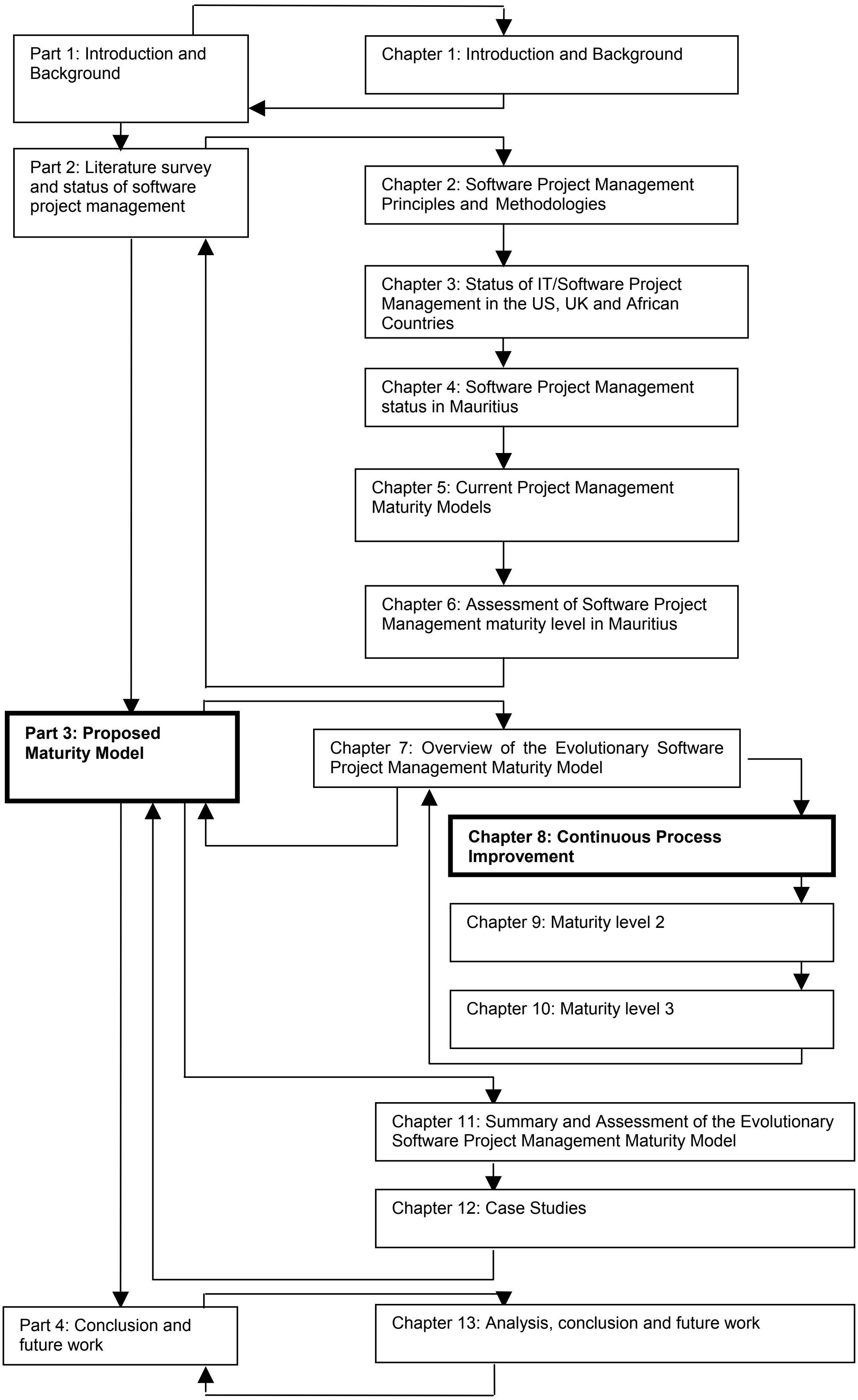


Figure 8.1: Chapter 8 in context within the overall thesis

8.1 Introduction

In chapter 7, the conceptual model of the $ESPM^3$ was introduced. Discussions were conducted on the various maturity levels (maturity levels 1, 2, 3 and the continuous process improvement group) and their key process areas (KPAs). The evolutionary nature of the KPAs was shown and described using the conceptual model.

In this chapter, the continuous process improvement group of KPAs is discussed in more detail. The continuous process improvement group of KPAs (M_c) comprises KPAs that are considered at all maturity levels. The KPAs identified are:

- Soft skills management KPA.
- Change management KPA.
- Software specific focus KPA.
- Integration management KPA.
- Environmental management KPA.

These KPAs were found to be applicable at all maturity levels of the $ESPM^3$. This means that the KPAs are continuously improved and they also influence other KPAs (of levels 2 and 3) by maintaining a continuous improvement process in order to adapt to the dynamic changes that occur within organisations.

Each KPA above is discussed from a Plan-Do-Check-Act (PDCA) perspective as shown below:

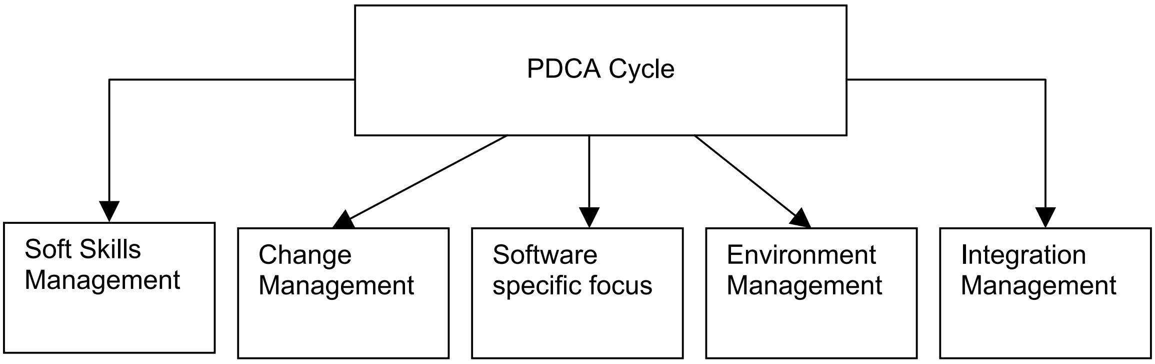


Figure 8.2: KPAs of the continuous process improvement group (M_c)

Although the cycle is preceded by the initiate process group and succeeded by the terminate process group (figure 7.10), discussions on these two are excluded as they are considered to mark a formal start and end of the PDCA cycle. The identification and classification of the KPAs resulted from a combined survey and interview carried out at 11 pertinent software development companies in Mauritius so as to gather information on the applicability of a software project management maturity model. Only 11 companies were considered as they represented the major companies involved in software development. The selected companies comprised, among others, organisations responsible for software development for the entire public service and most of the large private companies as well as international organisations.

8.1.1 Aim, objectives and structure of this chapter

In chapter 8, the continuous process improvement group of KPAs is described in greater detail than in chapter 7. The application of the Plan-Do-Check-Act cycle to the various KPAs is explained.

The remainder of the chapter is structured as follows:

Section 8.2 – In this section, a representation of the continuous process improvement group of KPAs is shown.

Section 8.3 to 8.7 – All the KPAs are described in detail in these sections and the PDCA cycle as applied to these KPAs is made clear.

Section 8.8 – The consideration for the social, political, economic and cultural aspects is highlighted in the conclusion section. The importance of soft skills management as a KPA is also noted.

8.2 Representation of continuous process improvement group of KPAs

As explained in the previous chapter, the KPAs within the continuous process improvement group are not tied to a specific maturity level but are applicable at all stages involving software development. In most maturity models, for example CMMI, continuous process improvement is considered at the last level. Organisations and their environments are continuously subject to change and according to the survey conducted it was agreed that continuous process improvement KPAs should operate at all maturity levels (figure 7.1). Harvey and Brown (2006) also recognised the need to consider continuous process improvement at all stages in its model for organisational change. Five KPAs were identified in this group (referred to as M_c) where M_c is the union of these KPAs (figure 8.1).

$$M_c = \bigcup_{k=1}^{p} KPA_{ck} \qquad (8.1)$$

with p the total number of KPAs in the continuous process improvement group. A specific KPA in M_c is identified by adding the subscript “k”, e.g. KPA_{ck} being the k^{th} KPA in M_c.

Hence,

- Soft skills management is identified as KPA_{c1}.
- Change management is identified as KPA_{c2}.
- Software specific focus is identified as KPA_{c3}.
- Integration management is identified as KPA_{c4}.
- Environmental management is identified as KPA_{c5}.

Given that there are five KPAs specific to Mauritius in M_c, the continuous process improvement group is instantiated as (figure 8.2):

$$M_c = \bigcup_{k=1}^{5} KPA_{ck} \qquad (8.2)$$

The continuous process improvement group of KPAs is applicable at all maturity levels. This is shown diagrammatically in figure 8.3.

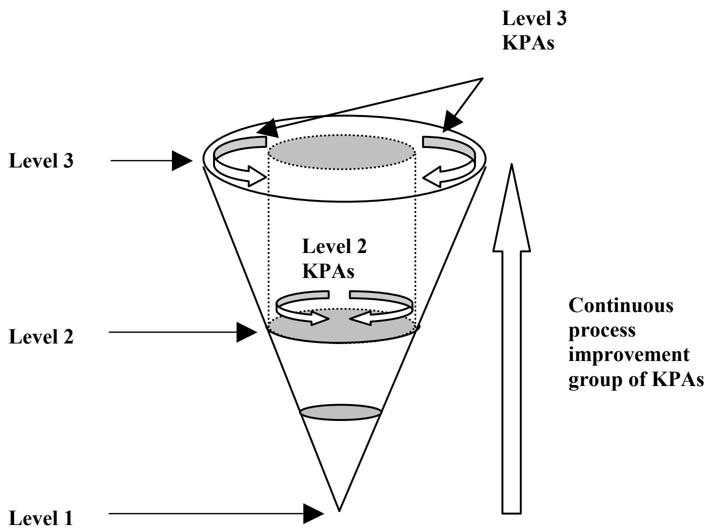


Figure 8.3: Continuous process improvement KPA within ESPM[3]

Therefore, the continuous process improvement group of KPAs spans the entire software development life cycle.

8.3 Soft Skills Management

According to Belzer (2004), hard skills are collectively described as a science comprising processes, tools and techniques that are applied to projects and are the main focus of many project management methodologies/models/standards. Extensive research has been carried out in the project management domain with greater focus on hard skills. So far, such skills can be learnt from formal courses. To this end, various organisations offer courses in project management methodologies for software specialists to learn the processes involved in managing software projects. These methodologies/models/standards are normally independent of any tools and techniques. The software specialist is free to choose among a range of tools and techniques.

In contrast, soft skills according to Belzer (2004) are collectively described as an art and are often ignored during software project management. They are often concerned with managing and working with people (Kirsch, 2004); they usually represent non-technical skills, and are typically acquired through experience. Several companies, like Mastek, Polaris and Sun Microsystems, are conscious about the importance of soft skills and have incorporated such skills into their training agenda (Arora, 2003).

Few projects fail because of inappropriate use of hard skills (Belzer, 2004). Instead, Belzer (2004) mentioned that projects fail due to the inability of project managers to communicate effectively, work within the organisation's culture, motivate the team, manage stakeholder expectations, understand business objectives and make clear and knowledgeable decisions. Soft skills required for software project management were identified by Sukhoo et al. (2005b). These skills (communication skills, team building, creativity, leadership, organisational culture dissemination, stress management, trustworthiness and conflict management) are discussed below:

8.3.1 Communication Skills

Belzer (2004) defined communication skills as the ability to convey ideas easily and clearly in order to ensure that the team moves towards a common goal. In addition, she mentions that "project managers must have the ability to convey complex ideas easily, clearly articulate what must be accomplished, keep the team moving towards a common goal, foster an environment that allows team members to communicate openly and honestly, admit their own mistakes without losing respect, negotiate, listen and facilitate". The invisibility of tasks associated with software development exacerbates the need for effective communication between all stakeholders. Therefore, effective communication skills can render visible those aspects of software development that are normally not visible to software users. These aspects include, among others, the progress of software projects, functionalities of some program modules and the complexity of some routines. In this way, misunderstandings or ambiguities can be cleared at an early stage. It is particularly important for the project manager at least to possess good communication skills in order to facilitate the flow of software requirements between the user and the development team. According to Sonnekus and Labuschagne (2004), even project teams need to bridge the gap between technical staff and business staff in order to better understand the needs of customers.

Very often, certain aspects such as language, learning styles, communication networks, preferred learning mode, memory access and retrieval systems, attention and retention, and various other factors affecting learning; information processing and communication are simply disregarded. Particularly, within the ambit of the IT domain, these aspects are minimised in favour of technology (Newstrom and Davis, 2001).

Many managers have now started to accept and appreciate the importance of NLP (Neuro Linguistic Programming) in the learning process, and in establishing effective communication patterns in, and between, organisations. One cannot deny the fact that the success of any project rests upon the judicious management of all information. Hence, the importance of a proper communication network and various related aspects should be carefully considered to ensure the success of projects where people are involved (Newstrom and Davis, 2001).

Throughout the software development process, communication of project information should be encouraged. Consideration has to be given to who needs what information, in what format and when. Two-way communication must be enhanced, thereby inviting important feedback to keep projects on track. In this way, the software project manager may also encourage the project team to be open for discussion and honest. The software project manager must ensure that team members are not intimidated by fear of repercussion (Bruce and Langdon, 2000).

The relationship that the project manager maintains with his/her team members determines the type of communication in terms of openness and honesty. Various means of communication like formal meetings, informal discussions and reports etc., must be encouraged. Also, the use of email may be an extremely useful timesaving device and the use thereof should be considered.

8.3.2 Team Building

Team building refers to the selection of members with an appropriate blend of skills so as to form a team capable of ensuring successful project completion. Team building helps establish group norms, group behaviour and desired attitudes as well as group identity. Software project managers should ensure that a full complement of people with a commensurate level and blend of skills and personalities are adequately represented on a project (Bruce and Langdon, 2000). The skills possessed by different individuals are often complementary and most of these skills are acquired through experience in contrast to hard skills that are learnt mostly through formal courses. Conflicting behaviour between team members should be detected and resolved as early as possible since software project management is typically a team activity, and groups of people working together formally may contribute towards the achievement of defined objectives. Teamwork is very significant to software project management as there is a clear need for considerable interaction between people carrying out different functions, but with a common purpose.

From senior management level down to lower levels, teamwork is important. Large and complex software products are developed from various modules by different team members. Teamwork has to be facilitated by the project manager so as to avoid any surprise during the final stages of the software development process. The project manager has to ensure and encourage individuals to join teams and create situations to promote effective teamwork. This can be done partly by making use of project review meetings to strengthen teamwork and help build team confidence. The collection of individual skills working towards a common goal constitutes a powerful force (Phillips, 2002). According to Gray and Larson (2000), the effect is that the whole is greater than the sum of the parts.

8.3.3 Creativity

Software development requires some creative skills in addition to hard skills. A particular software artefact developed by two different individuals is unlikely to be similar. The performance, size, complexity and usability may all differ for the software concerned. This is due to the fact that software development is influenced by the collective creativity of programmers and systems analysts. Since every software project has unique characteristics, each may call for a different combination of components, templates, tools and techniques (Belzer, 2004). The software project manager must also adapt to the needs of the project. Creativity can be promoted through the understanding of the personality of individual members, and acceptance of individual preferences (Santrock, 2000).

Creativity requires an appreciation of the intellectual abilities of the team members and the belief that each individual is unique. Opportunities provided by the software project leader to the team members ought to encourage creativity.

8.3.4 Leadership

Software project managers are required to lead a team of professionals and, therefore, they have to possess leadership qualities. According to Dobbins and Pettman (1997), leadership is the ability to motivate people to work towards achieving common goals and to make ordinary people display extraordinary performance. Leadership is related to a person's skills, abilities and degree of influence to inspire, motivate and direct people, making decisions and doing things that they would normally not have embarked on. Leadership is one of the functions of managers, hence software project managers are supposed to lead a software project team. Different styles of leadership, some better than others in certain circumstances, are adopted by software project managers. Good teamwork may result from consensus building between project managers and team members. A software project manager should be someone whom people want to seek out rather than avoid (Bruce and Langdon, 2000). According to Bruce and Langdon (2000), project managers display four leadership styles:

- *Dictatorial style*, which refers to the person's autocratic style and controlling character. It may prove to be helpful in a situation of crisis. The dictatorial style must be used with care and as infrequently as possible since it de-motivates the team members, thereby discouraging teamwork;
- *Analytical style*, where decisions are based on the gathering and analysis of facts. This style, which requires good analytical skills, is adapted when a project is running short of time and the correct decision must be taken quickly;
- *Opinion-seeking style* involves gathering team members' opinions in order to reach a decision. Team confidence is built and the project manager indicates that the views of the team are valued positively;
- *Democratic style*, which refers to team participation, encouragement and involvement in the decision-making process. It motivates the team and helps strengthen their commitment to the project.

8.3.5 Organisational culture dissemination

Project managers are required to contribute to organisational effectiveness by adapting to and disseminating organisational culture throughout the project team. An organisation with a dominant culture defines the boundary that limits the behaviour of its members; it conveys a sense of identity among its members; it facilitates the generation of commitment; it enhances internal social system stability and assists in keeping the organisation together; it also serves as a sense-making and control mechanism; and it guides and shapes the attitude and behaviour of the members of the organisation (Farago and Skyrme, 1995). It is, therefore, important for software project managers to understand the corporate culture, the organisational dynamics and the individuals they are dealing with (Belzer, 2004).

With the advent of Information Technology permeating all spheres of organisations, several characteristics that capture the essence of organisational culture have become more evident. Aspects such as innovation, risk taking, attention to detail, people orientation, team orientation, outcome orientation, stability, positive aggressiveness and other dimensions that directly or indirectly affect individual members should be given due consideration (Farago and Skyrme, 1995).

Organisational culture is particularly marked in Japanese companies. Employees tend to secure life-long employment and to work within the framework of the prevailing organisational culture (Ohashi High Technology Corp., 2007).

8.3.6 Stress Management

Hornby (1995) defines stress as pressure, tension or worry resulting from problems. Software project management is inherently a source of stress for project managers struggling to ensure that a particular software artefact is developed within time, cost and quality constraints. Stress management is necessary for the software project manager who has to perform his/her duties constantly under pressure and must not be overwhelmed to the extent that the team is also affected by his/her behaviour. A certain level of stress is important for the software project manager to perform well and this type of stress is called *eustress* (referred to as good stress). Eustress helps the project manager to achieve predefined targets. When stress becomes excessive, it can cause harm to the individual concerned or to the immediate surroundings. Beyond a certain level, it can cause *distress* (undesirable stress). According to Chapman (2001), stress sufferers can lose up to 50% of their aptitude to perform their tasks.

Software development is an area where constraints of deadlines, degrading quality of work and budget overruns can rapidly lead to a degeneration of the working environment. Distress may cause a deterioration of the situation. Maintaining a good level of eustress as early as possible and ensuring that it does not turn into distress can help to improve the progress of the project.

8.3.7 Trustworthiness

Trustworthiness is the value given to the belief that one can rely on the goodness, strength and ability of someone else (Hornby, 1995). Trust is the basic requirement that ensures the possible success of any human enterprise. Trust must prevail at all times between partners and between the manager and his/her subordinates. All relationships pertaining to the execution of the project under consideration ought to be based on trust.

Mutual trust must prevail throughout project execution between the software project manager and all of the stakeholders involved in the project. For example, a Director of a software development company has to trust the project manager for the efficient execution of a project rather than requesting the project manager to submit progress reports too frequently thus unnecessarily delaying the project. This applies also to software developers, who are too frequently requested to submit progress reports. A software project manager must also be able to convey that he/she can always be trusted to do what is right at the appropriate time to render the project successful and the client satisfied (Lussier, 1990).

8.3.8 Conflict Management

Conflicts are serious disagreements that emerge within the project team during the execution of a software project. It may arise due to disagreements over priorities, allocation of resources, quality of work (Gray and Larson, 2000) or other reasons. Conflicts may be desirable (especially, when they resolve ambiguities or when they raise concerns about a certain problem or pitfall) or can hinder the performance of a project (e.g. when both parties in conflict are wrong in the debate concerned and no one intends to reach a compromise). Desirable conflicts are often referred to as functional conflicts while undesirable conflicts are also called dysfunctional conflicts.

Conflict resolution skills are necessary for a software project manager to prevent behaviours of team members from degenerating into irreconcilable situations that may prohibit them from working together in a productive manner (Newstrom and Davis, 2001). Therefore, any conflicting behaviour has to be detected and resolved as soon as possible.

Dysfunctional conflict resolution is a challenging task for software project managers. It might not be identified until it becomes too late to overcome. The following strategies can be employed to manage conflict resolution:

- Mediation of conflict.
- Arbitration of conflict.
- Control of conflict.
- Accepting conflict.
- Elimination of conflict.

Soft skills management is carried out through the following inputs that are transformed into outputs:

- List of soft skills.
- Soft skills remedial action knowledge base.
- Historical data.

The transformation yields the following outputs:

- Satisfied team members.
- High performance and productivity.
- Lessons learnt to be used for future soft skills management.

The transformation of the inputs to the outputs is shown in figure 8.4:

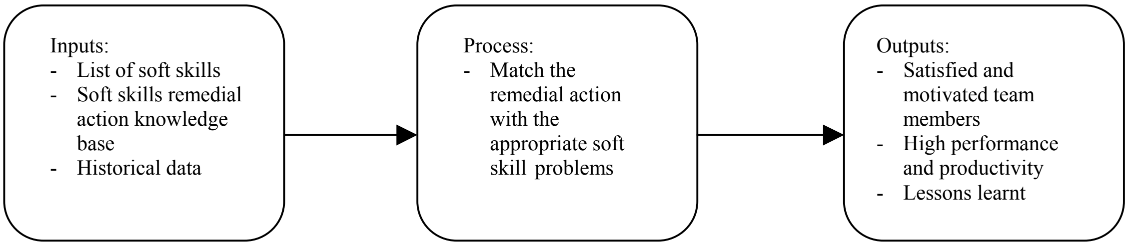


Figure 8.4 Transformation of inputs into outputs

8.3.9 The PDCA cycle for Soft Skills Management

Soft skills attributes of the software project manager have to be defined and then considered at the appropriate time during the different phases of a project. Soft skills management are mapped onto four process groups (plan, do, check and act) as per requirements of ISO 9001:2000 standard (Mauritian standard, 2001). This is shown in figure 8.5.

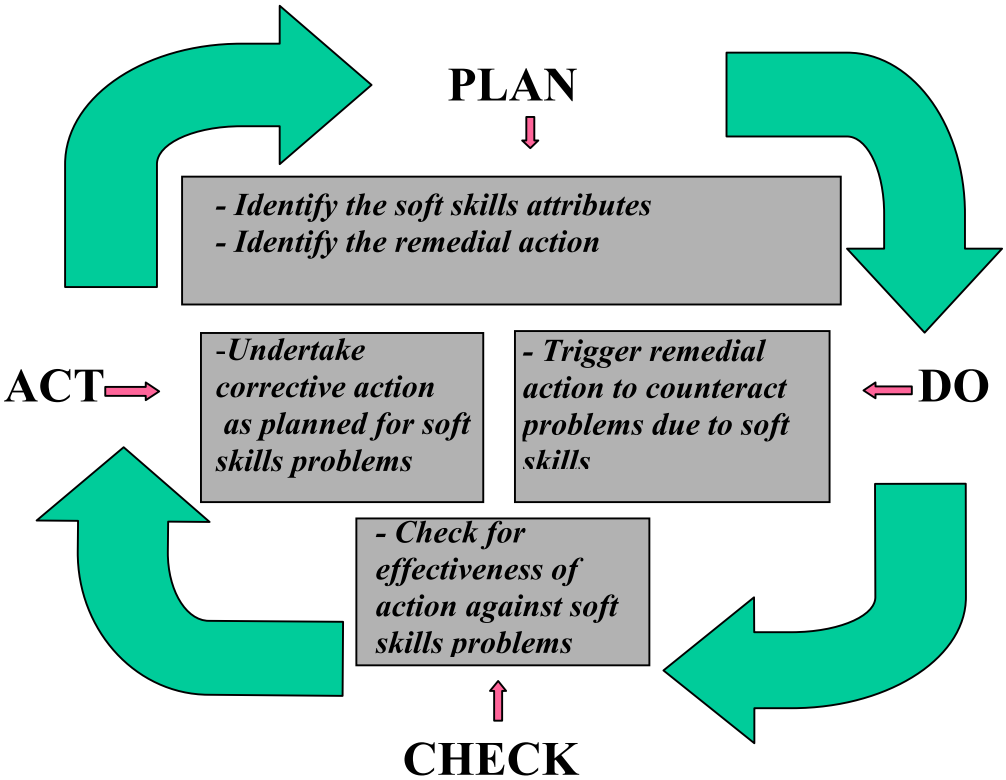


Figure 8.5: PDCA cycle for soft skills management

The project manager has to take into consideration the soft skills that need to be applied during the different phases of software project management.

Initially, this can be done in the form of a table listing the soft skills against the project management phases. However, with experience, the use of the respective soft skills will arise in a natural manner.

8.3.9.1 Plan: Soft Skills management

The objective of the planning process group for soft skills management is to deal efficiently with employees' problems involved in a project through soft skills consideration. It may be an extension of human resource management and emphasis is laid on soft skills identified.

It is important for documentation to be kept by the project manager so that such information is available for future projects. It is to be noted that soft skills management is to be carried out throughout the project development life cycle.

During the planning process group, the soft skills attributes that may be involved in executing the project are identified and the remedial actions are matched. The same list can be used across projects. Based on the discussion on soft skills above, table 8.1 is derived. The table can be filled by checking (that is, inserting a tick in the last column of the table) the appropriate action. The list of problems is not exhaustive. In practice, the list may also be produced or augmented through a brainstorming session.

Table 8.1: Soft skills plan

Problem requiring soft skills management	Soft skills attribute	Possible remedial actions	Appropriate action
Inadequate Communication	Communication skills	Honest discussion.	
		Listen to person concerned.	
		Negotiate.	
		Facilitate discussion.	
Inappropriate mix of skills	Team building	Swapping team members between teams.	
		Participation of a member from another team to provide services on a part time basis.	
Lack of user-friendliness of software	Creativity	Feedback from users.	
		Use of standards developed within the organisation.	
De-motivated staff	Leadership	Consensus building through listening and understanding the reasons for de-motivation.	
		Use of appropriate leadership style.	
		Provision of appropriate incentives and rewards.	
Opposing organisational culture	Organisational culture dissemination	Holding discussion to facilitate commitment.	
		Provide information on the need for the prevailing culture for efficiency in the organisation.	
Excessive stress originating from need to ensure delivery of projects within time, cost and quality constraints	Stress management	Regular feedback on the status of all aspects of a project and providing any corrective action.	
		Use of appropriate communication channel for reporting of problems pertaining to delivery of projects so that corrective action is taken.	
Lack of trust	Trustworthiness	Discussion, listening and understanding.	
		Consensus building to enhance trust.	
Conflict between team members	Conflict management	Mediation.	
		Arbitration.	
		Control.	
		Acceptance.	
		Elimination.	

8.3.9.2 Do: Soft Skills management

During this process group, appropriate remedial actions are taken for the relevant problems that occur as planned in the previous subsection and summarised as per table 8.1. Suppose during the execution of a project, a certain conflict arises between two team members and the project manager finds that he/she is able to persuade both persons to make concessions for the benefit of the project, then the first action is appropriate. The reasons for the action taken may be documented for future experience in dealing with a similar problem.

8.3.9.3 Check: Soft Skills management

Since the best-laid plans can go wrong, it is crucial to have an early-warning monitoring system (Bruce and Langdon, 2000). Regular meetings, progress reports and informal discussions can help in establishing a timely feedback system. In case problems occur once the project plan is executed, team members are encouraged to raise concerns (Bruce and Langdon, 2000). In this way, problems are, therefore, identified at an early stage rather than when a crisis situation has occurred.

It is even possible that new information or a change in the external environment may invalidate the plan. In that case, the plan is updated so as to reflect the prevailing situation.

The efficiency of the action taken is evaluated and documented. In the case of the example with conflict management considered above, the project manager needs to monitor the outcome of the solution. Any outcome that requires a corrective action is dealt with in the next process group (that is, Act process group) as per actions identified in table 8.1.

8.3.9.4 Act: Soft Skills management

As mentioned in the previous subsection, any necessary corrective action is taken by the Act process group. Any deviation from the desired effect is noted and immediate corrective actions are taken to prevent the project from being affected. Considering the same example of conflict, in case of continuous occurrence of the conflict, the project manager has to revisit the planning process group and select the appropriate solution. The solution may be an elimination of the conflict by removing one member from the team if the conflict is intolerable and hinders the progress of the project.

The PDCA cycle may continue as long as the problem has not been resolved. As soft skills management KPA is defined to be applicable at all maturity levels, the project manager must be constantly aware of problems that may occur at any stage during the development of the project.

8.4 Change Management

Changes are inevitable during the execution of a project and some circumstances may cause the team to feel de-motivated. It is, therefore, necessary to adjust the plan accordingly to accommodate changes. Some changes may be within the control of the software project manager (like shortening the time schedule) while others may not (like the transfer of one key team member to another project by senior management). Therefore, an assessment of the impact must be made with regards to changes to schedules, budgets and resources involved. Any alteration should be identified and evaluated before committing to a change request (Proehl, 2001).

One important change management issue is dealing with user resistance. Especially for software projects, changes are inseparable features as new developments in technology are brought about continually. Both users and team members have to be informed as early as possible. People’s concerns need to be taken into consideration and actions to dissipate their fears need to be taken as expeditiously as possible (Proehl, 2001). People are concerned, among others, with changes in work pattern, threat of job loss and losing authority. The project manager has to be able to reassure the users so as to obtain their cooperation.

In particular, user resistance can cause a software project to fail at the last stage (operational stage) after months or years of assiduous effort. Therefore, user resistance has to be taken care of from the outset of the project. One solution is to involve the users throughout the different software project phases (James, 1999) and they must be conscious that they belong to a team and a team spirit is required among all members. In this way their contributions should be valued for the successful implementation of the project.

Change management also concerns the management of change brought about by technology or by changing user requirements. The Information Technology sector is a particularly dynamic one where new versions of Database Management Systems, Operating Systems, Hardware and Software Development tools are regularly replacing older versions.

An organisation is subject to change throughout its existence in order to stay competitive. Therefore, this KPA is considered at all maturity levels. An organisation operating at maturity level 1 cannot wait till it reaches a higher level to address this issue. In that case, change management may be considered at an ad-hoc manner.

Using the ESPM[3], Change management is carried out using the following inputs:

- Nature of change (whether affecting the organisation, technology, staff, etc.).
- Historical data.
- Change management action knowledge base.

The change management process generates outputs in the form of:

- Successful management of changes.
- Improved organisation efficiency.
- Lessons learnt for future improvement of change management.

The transformation of the inputs to the outputs is shown in figure 8.6:

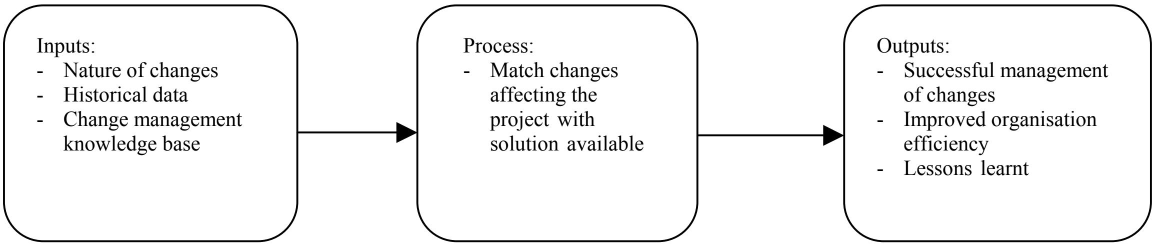


Figure 8.6: Transformation of inputs into outputs

Similar to soft skills management, change management is mapped onto four process groups, namely plan, do, check and act as shown in figure 8.7.

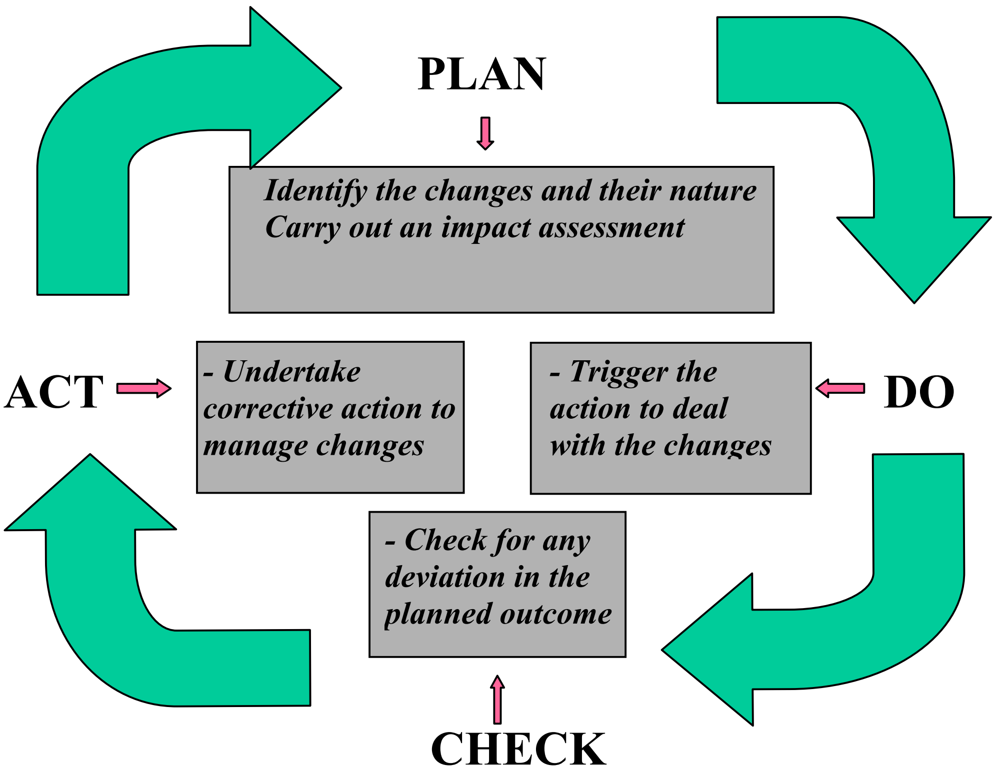


Figure 8.7: PDCA cycle for change management

8.4.1 Plan: Change Management

The objective of the planning process group for change management is to accommodate changes as they arise. Since changes are inevitable on projects, flexibility in their accommodation in the project plan is vital (Bruce and Langdon, 2000). An indicator for achieving the objective is the ability of the project manager to keep the schedule and budget as initially defined (which is the target of the project manager).

During the planning process group, all foreseen changes that may occur during the execution of the project are identified. Technology may change, user requirements may not be the same due to changes in legislation or organisational procedures and other KPAs will undergo continuous improvement. A list of potential changes that may have an impact on the project is prepared and means to address them are carefully identified. An assessment of the impact of the changes is carried out with respect to schedule, budget, quality constraints and resources initially planned for the project. Alternative solutions are worked out. It is proposed that a repository be constructed over time and used for all projects.

In the Information Technology sector, changes in technology occur regularly and project managers need to adapt software products under development to such changes. For example, a new Windows environment may not support an Oracle-based software. In such a situation, downgrading of the Windows Operating system is required pending the release of the appropriate patch file for the Oracle Relational Database Management System. Project managers must also keep abreast of forthcoming new technologies so as not to deliver a software product based on an outdated technology.

The preparation of an impact assessment plan may also help the project manager to identify and deal with changes that may occur during the execution of the project.

8.4.2 Do: Change Management

Any impact resulting from a change needs to be dealt with as planned. Changes normally affect the project plan in terms of schedule, budget, quality and resources. Therefore, changes need to be communicated to all parties concerned as early as possible. A well-planned change management process will help in identifying the solutions that are required to deal with problems arising from change in technology, staff resistance as well as change in user requirements.

Changes affect a project throughout its life cycle. Therefore, change management need to be considered at all phases in the development of a project. For example, user resistance may be encountered during the feasibility analysis phase as well as during the design phase where users may be reluctant to provide important information to project team members in order to resist changes that will have an impact on their jobs and authority.

8.4.3 Check: Change Management

The outcome of dealing with changes is monitored and the result is analysed during the check process group. The plan produced during the planning process group is updated if required and necessary documentation is kept for future projects.

This process group ensures that the impact of changes brought about in a project provides the necessary feedback for eventual intervention in case deviation from the plan is noticed. For example, changes in user requirements may impact on the schedule of the project. In such a situation, the user needs to be made aware of the changes in schedule. Similarly, senior management of the software development company needs also to be apprised of the situation as this may affect other projects as well.

8.4.4 Act: Change Management

Any deviation in the expected result is rectified during the Act process group. It must be ensured that the changes have been successfully dealt with as originally planned. Corrective actions need to be taken care of in the case of any deviation from the planned result. The PDCA cycle is applied to all the changes that arise. Any unsuccessful execution of change management may lead to the planning process group to be revisited so as to find an alternative solution. For example, the adoption of a new technology may not always lead to a successful project. There may be a need to revert temporarily to a well-tested older technology until the new technology is proved to be safe. For instance, a database application software artefact installed on a new Windows environment may not work and may require the operating system to be downgraded to an earlier version temporarily until the necessary patch files are released by the company supplying the Database Management System.

8.5 Software specific focus

In our investigations, software projects were found to be different from traditional engineering projects and were required to be managed in a different manner so as to take advantage of their strengths without disregarding their weaknesses (Sukhoo et al., 2004a). These strengths and weaknesses factors are:

- **Strengths of Software Projects (Sukhoo et al., 2005b)**
    - Flexibility.
      Flexibility refers to the ease at which a software product can be amended as compared to a traditional engineering artefact. Customisation of software is frequently performed in an attempt to shorten the development schedule. In contrast to the general flexibility of software products, traditional engineering artefacts do not lend themselves easily to modifications without structural impact.

- Mobility.

  Software can be transported at low or no cost at all since it can be rapidly transferred from one locale to another by utilising existing communication options. Traditional engineering artefacts on the other hand, very often, cannot be easily relocated from one physical location to another.

- Replication.

  Replication of the same software product is very common and this feature accounts for reduction in cost and duration of implementation. Traditional engineering projects, however, generally yield products that cannot be replicated as readily as is the case with software projects (except for some mass produced commercial items).

- Scalability.

  Software is scalable and can be adapted to new and emerging needs. A transaction processing system may be developed to record the day-to-day activities of an organisation. This system may then be used to develop a Management Information System for middle management to produce summarised structured reports.

  Scalability may not be the case for the majority of traditional engineering products. Another floor cannot be readily added to a building unless it has been planned beforehand, in which case supporting iron bars and columns of the necessary material of predetermined diameter must be carefully selected.

- Backup.

  Backup is an effective means to minimise data and software loss. The backup procedure in the software domain is carried out at minimum cost and effort while the backup of a traditional engineering product may prove to be too expensive to maintain. In the case of software and associated data loss resulting from hardware malfunction, deliberate corruption or natural disasters, recovery procedures can be initiated from the backup version. Clearly, this is not the case for traditional engineering artefacts.

- Reusability of components.

  Program modules used in one project can be reused a number of times in other projects, thus reducing development efforts resulting in time and cost savings. In the case of a traditional engineering product, however, a specific part cannot be removed from the said product and used in another product without usually rendering the original product non-operational.

- Possibility of using prototypes.

  Prototyping can be used to construct a final software product. This initiative reduces the effort and eventually the cost associated with the development of a software product. A traditional engineering prototype on the other hand is a representation that usually cannot easily be evolved to constitute the final product. For example, a model of a ship, a building or a car cannot readily be used to build the end product.

- Software evolution.

  A software project can evolve over time, resulting in new releases that outperform older versions. New releases tend to take advantage of prevailing technology where software evolution is then commonly used for improving software. Traditional engineering artefacts, like bridges and buildings are not necessarily built with an attempt to evolve them or to take advantage of new technologies to improve them.

- **Weaknesses of Software Projects (Sukhoo et al., 2005b)**

  - Invisibility.

    In the case of the construction of a physical artefact such as a bridge, the progress of work can actually be seen. One major weakness is the invisibility or poor visibility of progress during the development of a software project. The user may not be able to conceptualise the software until it is complete. If the user is not satisfied, a great deal of effort may have already been expended, ultimately resulting in project failure.

In contrast to the invisibility usually associated with the development of software products, traditional engineering product development is normally visible as it lends itself to observation of tasks while they are carried out.

- Complexity.

  Software artefacts are inherently more complex than traditional engineering artefacts. At the inception of a project, inaccurate or ambiguous information gathered from the users may constitute a major threat to subsequent stages of the software development process. A logical error may propagate throughout the development process and can go undetected until the last stages. In traditional engineering projects, complexity is relatively lower owing to better visibility of progress and usually well-understood requirements from the onset.
- Difficulty regarding estimation.

  Estimation of the software production process in terms of cost and duration is perceived as a difficult task. An inaccurate estimation may lead to cost overrun, failure to meet deadline or degradation in quality. Traditional engineering artefact development has been in operation for eons and consequently the visibility of the amount of work concerned, price of components, and existing well-defined complexity measures, may account for relatively easy negotiations regarding cost and time schedules between the user and the developer.
- Dynamic nature.

  The dynamic nature of software becomes a limitation in certain cases where new versions are no longer compatible or suitable to run on older versions of operating systems software or hardware. Traditional engineering products are not affected so much and at the same frequency as software products.

- Intangibility.

  The intangibility feature associated with software makes it difficult for potential clients to purchase a certain product. The tangibility associated with a traditional engineering product, is evident throughout the construction process. The client may watch the progress of the project on site and even communicate any issues of dissatisfaction to the developer immediately. This may enhance client satisfaction at the closure of the project.
- Regular upgrades.

  Regular upgrades improve the quality, performance and efficiency of software, but in the long run becomes expensive. Sometimes older versions of software are no longer maintained or the maintenance service is offered at a higher cost. The user is forced to bear this cost to stay competitive. Traditional engineering products do not require frequent upgrades, as is the case with software products. Once a building is completed, it is planned to be in that state for years. One can argue with a degree of certainty that additional floors will not be added annually to the building.
- Bugs.

  It is often observed that software suffers from bugs and flaws from the time it is released, requiring the software developer to be called upon to correct them. Sometimes, this affects the confidence of a user towards a software developer. Flaws in traditional engineering products are not so common after release due to a number of factors. These factors include, but are not limited to, the better visibility during the development process, an acceptable level of complexity and existing formal methods to assist the development process.
- Difficulty to add people to a delayed project.

  It is difficult to meet project deadlines by adding new people towards the end of the development cycle of a software project. It is often easier to add new people on a traditional engineering project that is behind schedule. For example in the construction of a bridge or a building, adding more people will generally speed up the construction process. However, the number of additional people must be carefully chosen so as not to impact on management overheads.

- Training.

  Software projects, most often if not always, need training sessions to be incorporated to use the end product (software). In the case of traditional engineering products, this may often not be the case. Generally, the user does not require extensive training to use a bridge, building or road.

8.5.1 The PDCA cycle for Software specific focus

Inputs are identified and the software specific process to transform them into outputs is developed. The following inputs are considered:

- Identified strengths and weaknesses.
- Appropriate actions to deal with corresponding strengths and weaknesses.
- Historical data.

The outputs of the software specific process generated are:

- Specific management techniques for software projects.
- Improved productivity.

The transformation of the inputs to the outputs is shown in figure 8.8:

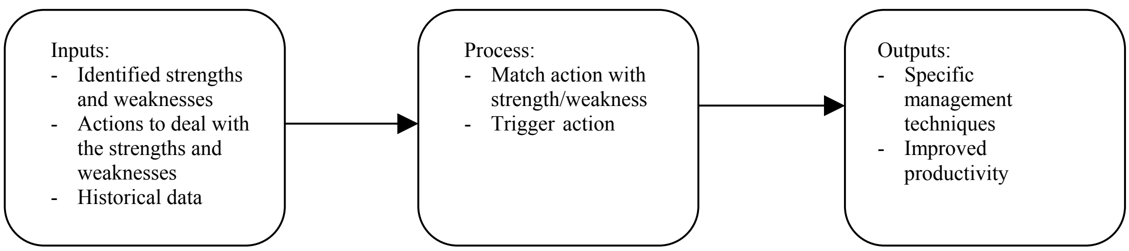


Figure 8.8: Transformation of inputs into outputs

The strengths and weaknesses are mapped onto four process groups, namely plan, do, check and act as shown in figure 8.9.

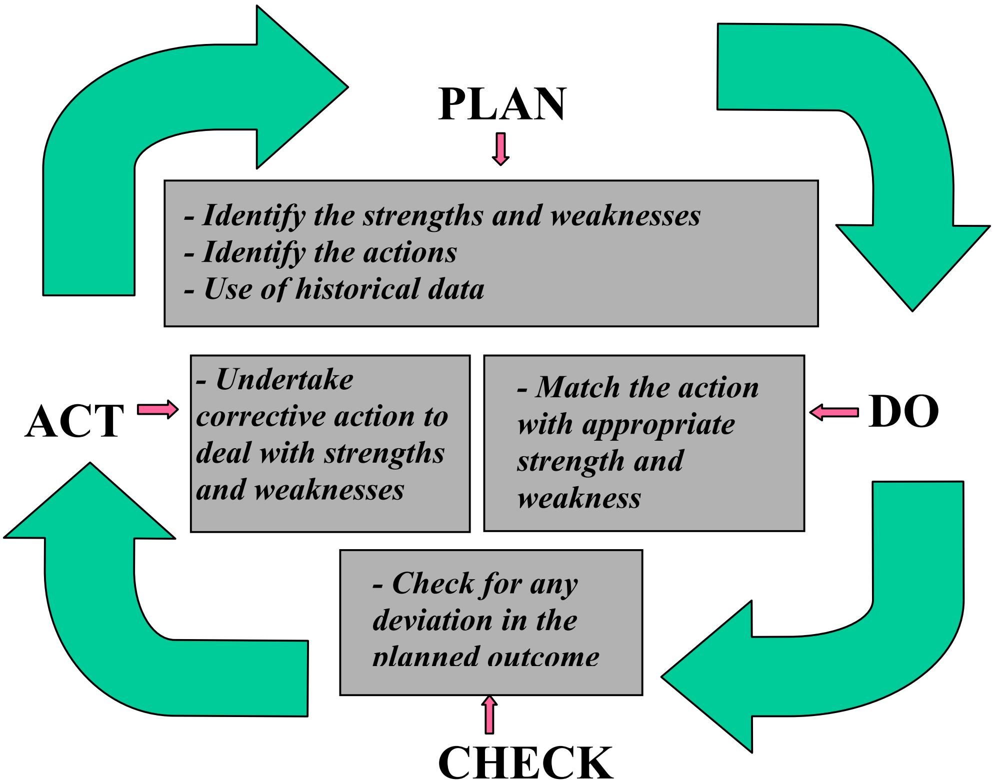


Figure 8.9: PDCA cycle for software specific focus

8.5.1.1 Plan: Software specific focus

The objective of the planning process group for the software specific focus is to consider the specificities of a software project based on the strengths and weaknesses identified above in the management of the project.

For this process group, all the strengths and weaknesses are identified and listed together with the actions to be taken whenever required. Historical data is also identified. For example, one of the strengths of a software project is the reusability of components. The project manager needs to be aware of the availability of reusable components. This initiative will certainly have a positive impact on the development time and may reduce the overall time span of the project. However, software strengths may reduce the project development time while weaknesses may extend the development time. This may also have an impact on the cost as well. Therefore, the combined effect of the software strengths and weaknesses has to be evaluated in order to manage the project and deliver it within time, cost and quality constraints.

8.5.1.2 Do: Software specific focus

The plan prepared in the Plan process group is executed while the project is carried out. Actions matching the respective strengths and weaknesses of software projects are identified and appropriate actions are taken. For example, components that are reusable and are available in a library are identified in the plan. During the development of the software, these components are integrated into the development software instead of building them from scratch. Similarly, during a testing phase, a test plan developed during the planning phase is used to minimise the risk of bugs affecting the project.

8.5.1.3 Check: Software specific focus

The outcome of the previous action is monitored and assessed. For example, the impact of including reusable software components may be so beneficial to the project that users are left satisfied with respect to development time throughout the project life cycle especially when some modules can be delivered prior to the completion of the entire project. This may be the case when the users are provided with program modules that may allow data capture with the necessary validation so that the data are ready when the entire project is completed.

8.5.1.4 Act: Software specific focus

This process group takes into consideration the output of the previous process group (Check process group) as a feedback to determine any corrective action deemed necessary. Any deviation that causes a negative impact is rectified as early as possible.

For example, complexity in software development may be reduced by means of a Work Breakdown Structure and subsequent development of simpler individual modules. The development of these modules is assigned to more than one programmer and their tasks are carefully planned in advance. An indicator to the success of the process group is the reduction in time for the development process. In case the indicator shows that there is no significant reduction in time as planned, the intervention of the project manager is required to investigate into the problem.

8.6 Environmental management

The Environmental KPA includes the cultural, economic, social, political and legal aspects that are often ignored in a software development environment. According to Muriithi and Crawford (2003), these factors have to be taken into consideration while managing projects.

According to Hofstede, Namenwirth and Weber, as cited in Hills (2005), culture is defined as a system of values and norms that are shared among a group of people and that when taken together, constitute a design for living. Mauritius is characterised by a multi-cultural society. The development of ICT into the fifth pillar of the Mauritian economy has initiated actions on the part of its Government to attract foreigners to invest in this sector. The result is that many foreign companies have started operations in Mauritius. These companies are compelled by law to employ a certain percentage of the Mauritian workforce in addition to foreign workers. Project managers need to motivate workers with different cultural backgrounds to interact constructively for the benefit of projects, and eventually the entire organisation.

8.6.1 Social and cultural aspects pertinent to Mauritius

The social aspect divides the Mauritian population into different stratas forming heterogeneous groups. Family ties are strong within the African communities, as pointed out by Muriithi and Crawford (2003). This is also true for the Mauritian community, where individuals have an obligation towards their families as Mauritius belongs to African developing countries (Muriithi and

Crawford, 2003). The characteristics of the social conditions are also summarised in chapter 3.

Muriithi and Crawford (2003) found that the political institutions of African countries are, in general, weak and unstable. Mauritius as a developing country has a parliamentary democracy based on the Westminsterian model of the United Kingdom since its independence in 1968. The politics of Mauritius takes place in a framework of a parliamentary representative democratic republic, whereby the Prime Minister is the head of the government, and of a pluriform multi-party (Politics of Mauritius, 2006). Executive power is exercised by the government. Legislative power is vested in both the government and the National Assembly.

According to Muriithi and Crawford (2003):

- Political interference occurs in African developing countries (including Mauritius) such that priority is given to projects that bring political benefits, e.g. employment opportunities or facilities to the nation. Such projects may often succeed at the expense of others, especially with the limited availability of resources;
- Managerial success requires political skills and, therefore, the ability of managers to maintain political connections by ensuring access to politicians is important;
- Projects that seek questioning of existing systems need to be dealt with in a politically "sensitive" language by project managers and project teams.

The legal framework in a country is a factor that may affect software development. Software piracy is tolerated in some countries and this has a negative effect on the creativity and motivation of software developers. In order for software development to become a successful industry, it is imperative that the intellectual property of software developers be protected. Mauritius has recognised the need for adequate safeguards by enacting the following (Ministry of IT & Telecommunications, 2008):

- Data Protection Act 2004;
- Computer Misuse and Cybercrime Act 2003;
- Information and Communication Technologies Act 2001;
- Electronic Transactions Act 2000;
- The Information Technology (Miscellaneous Provisions) Act 1998.

An insight into these Acts is important for project managers and project teams so that they are aware of the necessary safeguards for them as well as to prevent them from transgressing any of them.

8.6.2 The PDCA cycle for Environmental management

In managing the environmental factors, the following inputs are transformed into outputs:

- List of environmental factors to be addressed.
- List of ways to deal with the factors.
- Historical data.

The transformation yields the following outputs:

- Satisfied team members.
- Safe software development environment.
- Lessons learnt.

The transformation of the inputs to the outputs is shown in figure 8.10:

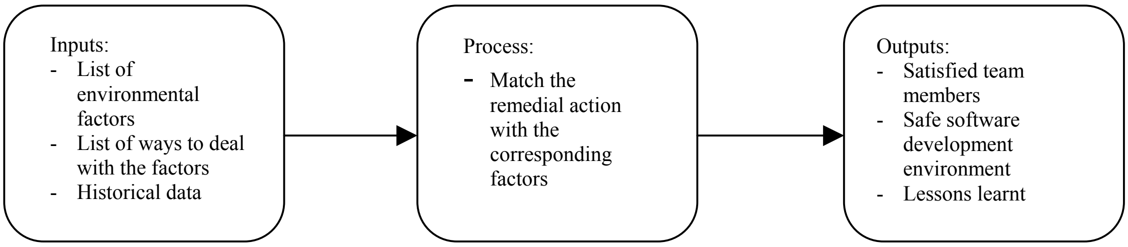


Figure 8.10: Transformation of inputs into outputs

In the ESPM[3], the environmental factors are taken into consideration. A list of factors is prepared and actions to be taken are defined. These factors are mapped onto four process groups, namely plan, do, check and act as shown in figure 8.11.

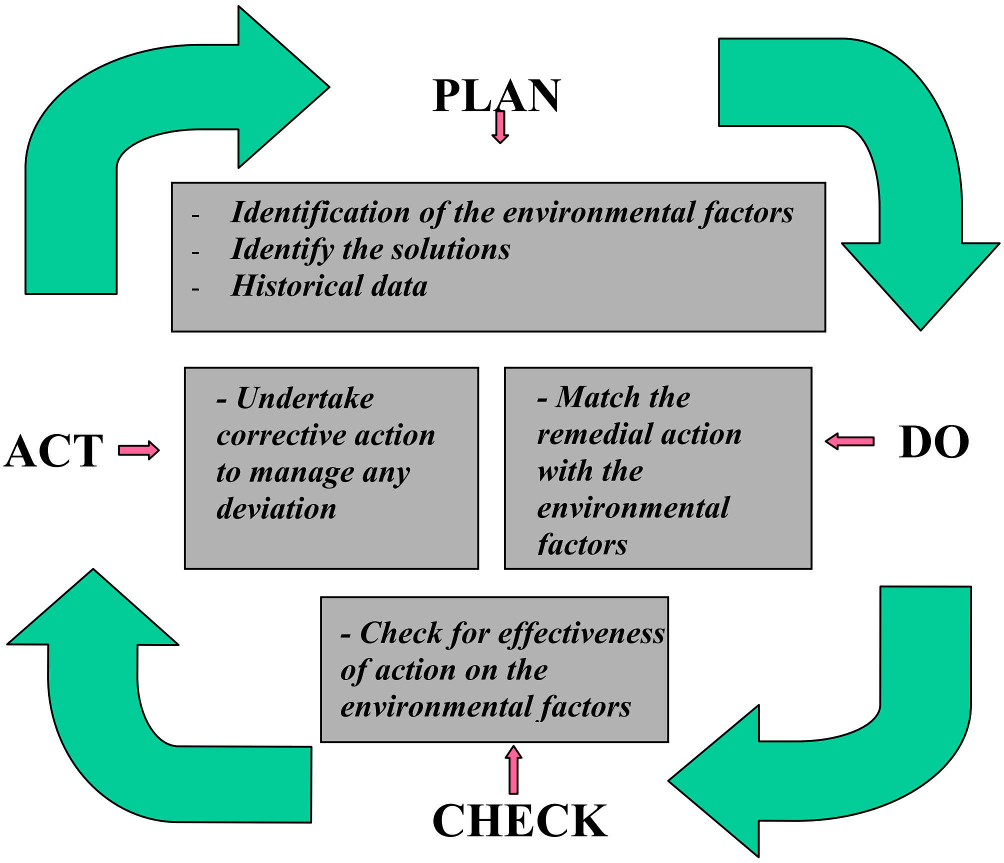


Figure 8.11: PDCA cycle for Environment management

8.6.2.1 Plan: Environmental management

The objective of the planning process group for environmental management is to consider the social, cultural, political and economic context in the management of the project. An indicator for achieving the social and political benefits is the reduction in time for software development. In the case of the cultural context, working with people of different cultures (locals and foreigners) may pose some challenges as far as interactions are concerned.

Many software development companies may have difficulties to retain skilled workers. The mobility of workers is exacerbated by the need to earn higher salaries due to rising inflation in the country. In addition, workers tend to seek better job security by joining larger companies. Several indicators are required in the case of cultural and economic context and they include maintaining the planned time schedule, cost of the project and quality standard.

The legal aspect must not be ignored, especially with the number of acts that have been proclaimed in Mauritius, e.g. the Data Protection Act (Ministry of IT & Telecommunications, 2008) that prescribes the way in which live data has to be processed for testing purposes. Naturally, all the legislations relating to the project need to be listed and to ensure regulatory compliance.

For this process group, the relevant factors namely cultural, economic, social, political and legal factors are listed and the relevant actions are considered in order to deal with issues that may arise. For example, a project that is deemed to provide political benefits needs to be closely monitored and any deviation from the plan needs to be addressed promptly. Such a project may also have a social impact where the public will have to be constantly apprised of the status by the media. For example in Mauritius, for the financial year ending June 2007, a total of 50 online services was announced through the local press to be available on the Internet. The project team managed to develop 48 such services in less than 6 months and they are currently available on the Internet (NCB, 2008). Such applications had to be given high priority over other, also probably, important applications.

The use of historical data compensates for experience in dealing with these factors.

8.6.2.2 Do: Environmental management

A plan prepared in the plan process group, with the environmental factors and ways to deal with them in the case of problems, is executed throughout the project life cycle. Actions matching the problems encountered are carried out.

For example, a project that is considered as a priority in terms of the political benefits that can be derived must be completed with extreme care. The project manager has to ensure that resources are available and assignment of tasks is well planned. In the event that a key staff member becomes unavailable, his/her replacement must be made immediately. The overall project plan must be strictly adhered to.

In this process group, the remedial action is matched with the respective problem in order to ensure that the project is delivered according to the schedule. In the example above, the problem occurring due to unavailability of a staff member is matched with the supply of a replacement so that the project is unaffected.

8.6.2.3 Check: Environmental management

The previous process group generates an output that needs to be verified against the plan produced during the plan process group. For example, the efficiency of the replacement staff is verified so that it matches the efficiency of the unavailable staff. The absence of a key staff member will have an impact on the delivery schedule of the project unless the plan takes into account alternative solutions that counteract the effect.

The efficiency of the action taken is evaluated for eventual corrective measures to be taken care of in the next phase.

8.6.2.4 Act: Environmental management

The feedback from the previous process group is gathered so that the appropriate action is determined. In the case of the replacement staff, his/her efficiency is a determining factor in order to ensure the successful delivery of the project. At an early stage, any discrepancy in the result is handled by the project manager. Any corrective action aims at the replacement staff or adding more staff on the project.

In the case of a project that is expected to provide a social benefit, a similar strategy as above is aimed at. Citizens expect that the project is delivered without any delay. In Mauritius, software development for providing an income support to vulnerable groups (e.g., the poor) was considered with great care, otherwise it would have eventually led to a political impact.

8.7 Integration management

The various KPAs cannot be dealt with in isolation, since all the processes involved have to be integrated in a well-coordinated manner. In the ESPM[3], one KPA called the integration management KPA ties all the other KPAs together so as to ensure that a common goal is attained.

The following inputs are considered in integration management:

- Other KPAs.
- Constraints.

The outputs generated are:

- Project results (in terms of scope, time, cost and quality).
- Lessons learnt.

The transformation of the inputs to the outputs is shown in figure 8.12:

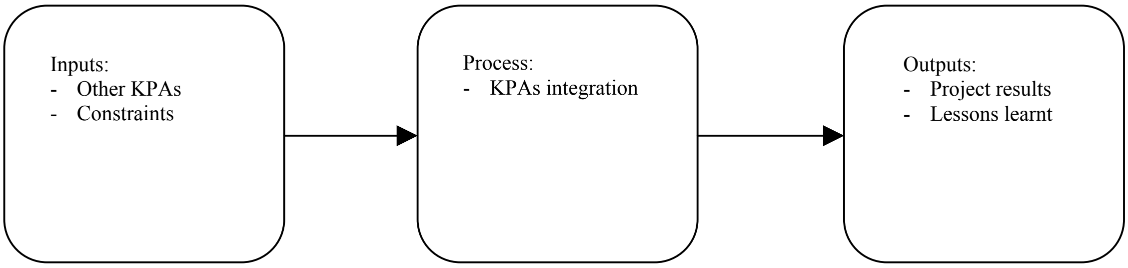


Figure 8.12 Transformation of inputs into outputs

The mapping of integration management onto the four process groups, namely plan, do, check and act as shown in figure 8.13.

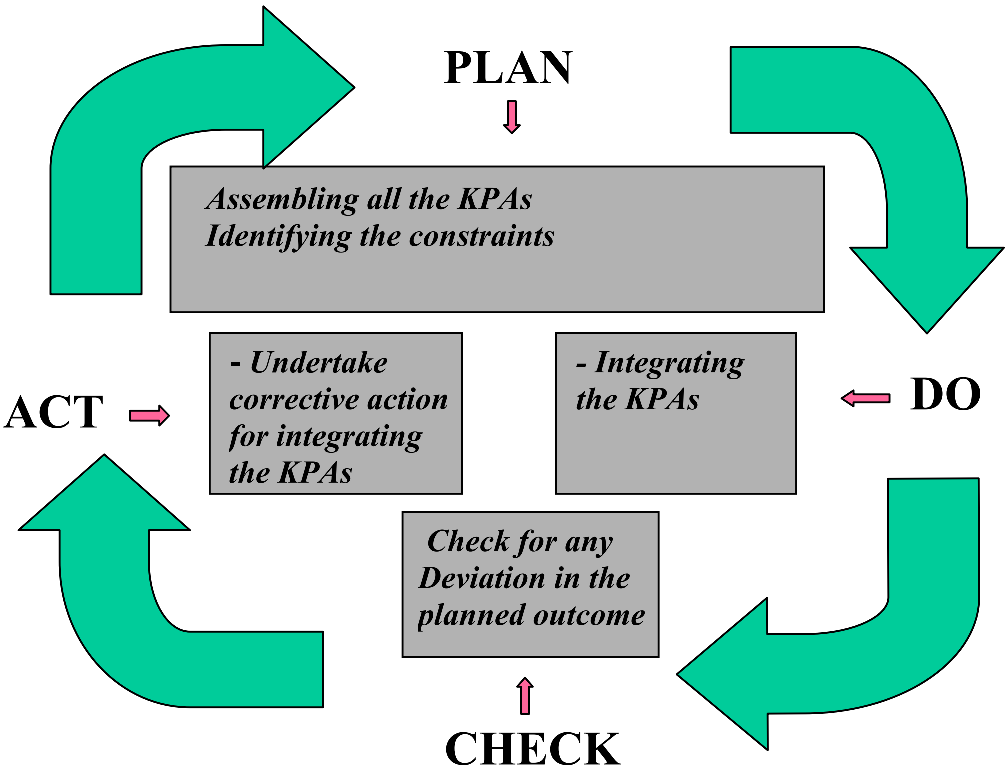


Figure 8.13 PDCA cycle for Integration management

8.7.1 Plan: Integration management

The objective of the planning process group for integration management is to bind all KPAs together so that they are considered in a consistent and coherent manner. For example, time management and budget management are related. An extension of the schedule due to inaccurate estimation results in an increase in the cost of the project. This cost may be passed on to the user or in the case of a fixed cost project, the software development organisation may have to bear this cost and thus accept a reduction in profit. The successful integration of KPAs may be measured in terms of time, cost and quality of the software product under development. Therefore, these parameters must be rendered visible as early as possible through the use of software tools.

In this process group, the outputs of other planning process groups are used as inputs. An overall project plan is developed. It includes information such as the objective of the project, the project management approach used, a Work Breakdown Structure, the time schedule, cost estimates, key resources, milestones, potential risks and quality standard. The activities involved and their interrelationships are highlighted.

8.7.2 Do: Integration management

In this process group, the project plan is translated into action. The project plan is executed by making use of the resources identified. The activities identified in the project plan are carried out. Each activity consumes time and part of the budget. Integration management ensures that the required software product is developed within agreed time, budget and quality constraints. It makes use of all the KPAs developed so far to build a software product.

8.7.3 Check: Integration management

The execution of the project plan involves several activities. The actual output of each activity is verified against the planned output. For example, the actual duration of activities is compared with the planned duration. This information is normally available graphically by the use of a Gantt chart. The actual costs of the different activities are also compared with the planned costs. As far as possible, software tools like Microsoft Project, Primavera or others have to be used to enhance the visibility of activities.

8.7.4 Act: Integration management

The comparison of the actual output against the planned output in the previous process group determines the deviation in the project plan. During the "Act" process group, appropriate actions are taken in the case of any deviation from the plan. For example, a longer duration than planned for an activity necessitates intervention of the project manager. The cause of the deviation is determined and corrective action is taken.

8.8 Conclusion

In chapter 7, an overview of $ESPM^3$ was presented, whereby the different groups (continuous process improvement, level 2 and level 3) of KPAs were introduced. In this chapter, the role of the continuous process improvement group of KPAs within the $ESPM^3$ was discussed. These KPAs were found to be essential for all maturity levels for software project management.

A KPA for soft skills management within the continuous process improvement group ensures that effective communication, team building, conflicts, creativity, leadership, organisation culture dissemination, stress and trustworthiness are handled and dealt with successfully. Researchers like Belzer (2004) and Arora (2003) emphasised on the need of soft skills in project management. Therefore, project managers need to be aware of the extent to which soft skills can help in managing software projects and apply them at all relevant stages during software development in order to motivate the project team.

Changes in user behaviours, user requirements, technology, organisational procedures and legislations are among the most pertinent factors that lead to difficulties as regard to management of software projects. Therefore, change management is considered in this group of KPAs so as to ensure that software projects are managed proactively.

Software projects have been quoted as complex and with lack of visibility features as compared to traditional engineering projects. Hughes and Cotterell

(2006) are among some researchers who concur with this fact. Software projects have their own inherent strengths and weaknesses that need to be taken into consideration. Therefore, the ESPM^3 includes one KPA with a software specific focus in order to take advantage of software strengths and at the same time mitigate or eliminate the associated weaknesses.

Software project management may differ from country to country due to cultural, social, political, economic and legal variations. It has been noted that often projects in developing countries are managed by ignoring these factors. Muriithi and Crawford (2003) raised concern in their research conducted in African developing countries. Stuckenbruck and Zomorrodian (1987) also revealed that project management methodologies/models/standards of Western origin are not necessarily universally valid. In South Africa, Sonnekus and Labuschagne (2004) highlighted the fact that communication is responsible for most failures and challenges as regard to IT projects. Communication in South Africa is frequently not based in a person's first language (Van der Poll and Van der Poll, 2007). This is true for most African countries, including Mauritius. The proposed ESPM^3, therefore, includes a KPA for environmental management so that these factors are not left out.

Finally, the continuous process improvement group includes a KPA to allow all other KPAs to be considered in a coherent manner rather than treating each KPA in isolation. The integration management KPA performs this task of binding all KPAs together in order to achieve a common goal of driving a project successfully.

The project manager will, therefore, consider soft skills management, change management, software specific factors, and environmental factors while developing the time management, cost management and quality management KPA and integrate all of them in a coherent manner until the KPAs of maturity level 2 attain maturity.

ESPM^3 essentially consists of three main groups of KPAs – continuous process improvement group, level 2 KPAs and level 3 KPAs. This chapter presented a comprehensive explanation of the continuous improvement group of KPAs. In Chapter 9 the KPAs for maturity level 2 will be discussed, followed by the level 3 KPAs in chapter 10.

Chapter 9

Maturity Level 2

Table of contents

9.0 Chapter layout

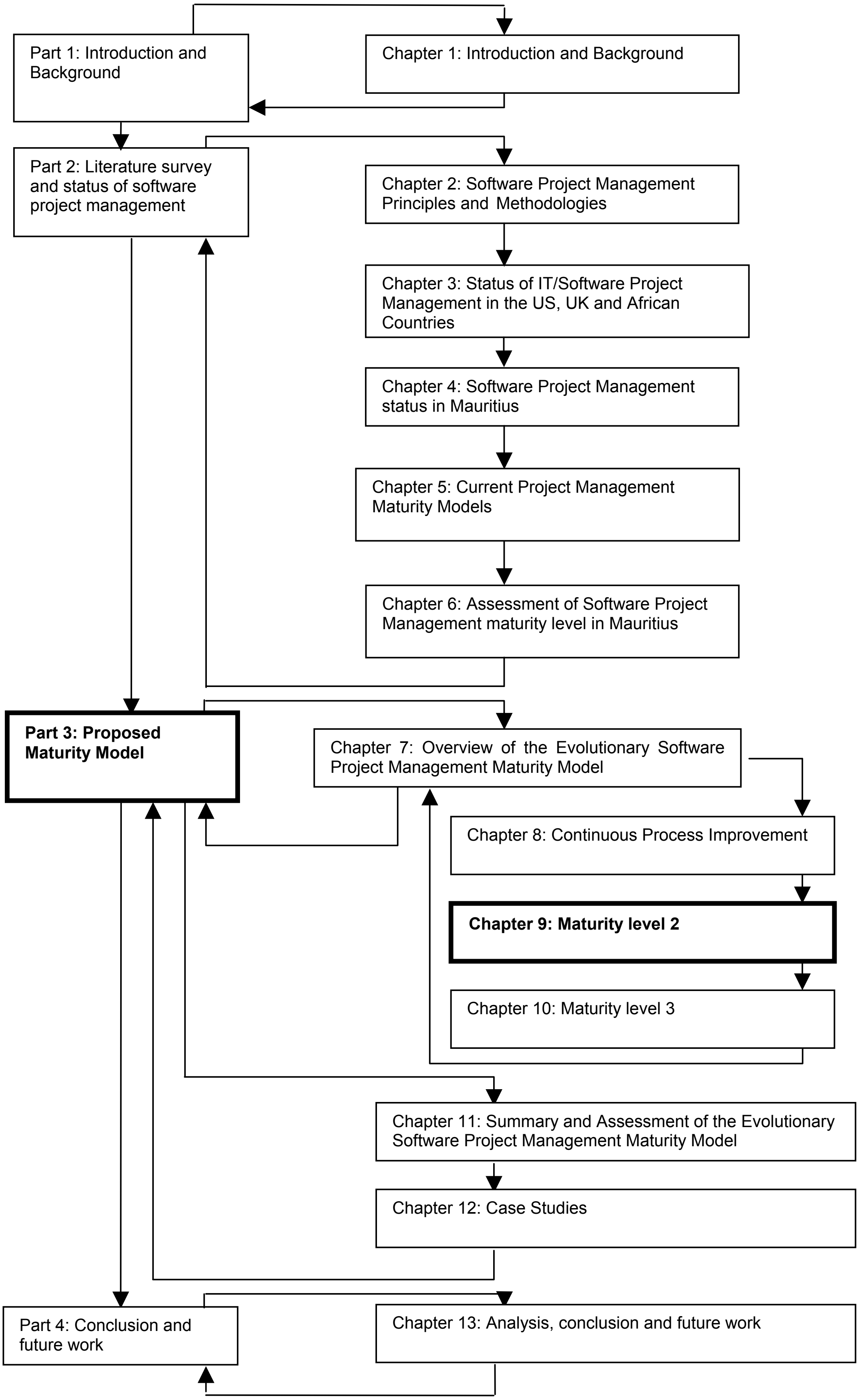


Figure 9.1: Chapter 9 in context within the overall thesis

9.1 Introduction

In chapter 8, the KPAs applicable at all maturity levels were discussed. Essentially, they perform a facilitating function by supporting timely delivery of projects within budget and quality standard. As such, the soft skills management KPA helps in enhancing communication, building a team spirit, managing stress, etc. Given that a software project is under development, the strengths and weaknesses of software projects are taken into account. Similarly, the change management KPA and the integration management KPA is applied at all maturity levels as changes may occur at any stage during software development and all the KPAs have to be applied in an integrated manner rather than in isolation.

This chapter brings into perspective the KPAs that provide a core function of project management. Maturity level 2 of the ESPM[3] is described as the basic project management level. It comprises three KPAs, namely:

- Time management.
- Cost management.
- Quality management.

Software projects suffer from a lack of direct visible features regarding duration, progress and quality as compared to most traditional engineering projects (Sukhoo et al., 2005b). Prior to the delivery of a software product, the user may not know in advance whether or not the software will satisfy his/her requirements satisfactorily. Therefore, it is important through the use of appropriate tools and techniques to render visible the progress of software projects. All the three variables (time, cost and quality) are carefully controlled so as to improve the success rate of software projects. These variables have been depicted as the apex of a triangle, sometimes called the golden triangle of cost, time and quality (Gardiner and Stewart, 2000), as shown in figure 9.2:

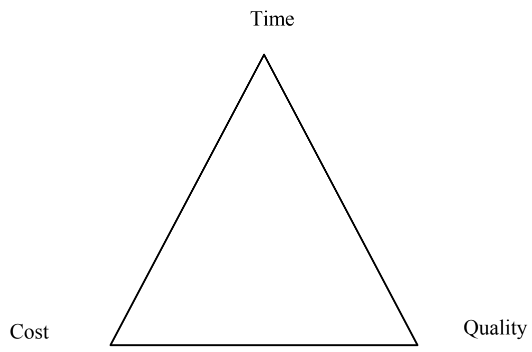


Figure 9.2: Golden triangle of cost, time and quality

Sometimes, project scope is used as a fourth variable as in the case of PMBOK®. In the case of the proposed model (ESPM^3), project scope is taken care of in project quality.

This chapter elaborates on the three key process areas, the tools and techniques available to make an attempt to deliver software projects on time, within cost and as per initially set quality standard. Suggestions are also made with regards to the tools and techniques that Mauritian software organisations can consider.

9.1.1 Aim, objectives and structure of this chapter

In chapter 9, the maturity level 2 KPAs are described in greater detail than in chapter 7. The application of the Plan-Do-Check-Act cycle to the various KPAs is also explained.

The remainder of the chapter is structured as follows:

Section 9.2 – In this section, a conceptual representation of maturity level 2 is shown diagrammatically and explained.

Section 9.3 to 9.5 – All the KPAs are described in detail in these sections and the PDCA cycle as applied to these KPAs is made clear.

Section 9.6 – In this section, the evolution from maturity level 1 to level 2 is explained.

Section 9.7 – The conclusion aims at showing the importance of the various KPAs as discussed in previous sections.

9.2 KPAs at Maturity level 2

In addition to the KPAs specific for maturity level 2, the continuous process improvement group of KPAs provides a facilitating function to the core function KPAs. Soft skills management ensures that a conducive environment prevails during software development. Change management permits changes in requirements to be accepted with minimum disturbances to the schedule, cost and quality. Strengths associated with software projects are taken into consideration (for example, the use of evolutionary prototyping can clear up misunderstandings at an early stage during the software development process) and as far as possible, inherent weaknesses are mitigated (for example, invisible features like progress of the software development process can be rendered visible through the use of appropriate tools such as Gantt charts and the Critical Path Method). Environmental management, like responding to political and clients' pressures, needs to be addressed at the same time when the KPAs responsible for the core functions (Time, Cost and Quality) are being considered. In addition, the integration management KPA ensures that all other KPAs are executed in a coherent manner thus allowing software development to proceed in a streamlined manner.

The KPAs at maturity level 2 (also discussed in chapter 7) is revisited below:

Maturity level 2, $M_{2,}$ is the union of all KPAs at level 1 (M_1), all KPAs at the current level and the continuous process improvement group of KPAs (M_c). Given that $M_1 = Ø$,

$$M_2 = M_1 \cup \bigcup_{j=1}^{n} KPA_{2j} \cup M_c$$

A total of 3 specific KPAs are identified at maturity level 2 and, therefore, n=3 in the above equation. Thus,

$$M_2 = \bigcup_{j=1}^{3} KPA_{2j} \cup M_c$$

The KPAs are as follows:

- KPA_{21} = Time management
- KPA_{22} = Cost management
- KPA_{23} = Quality management

The KPAs are developed till maturity at level 2 is reached as shown in figure 9.3.

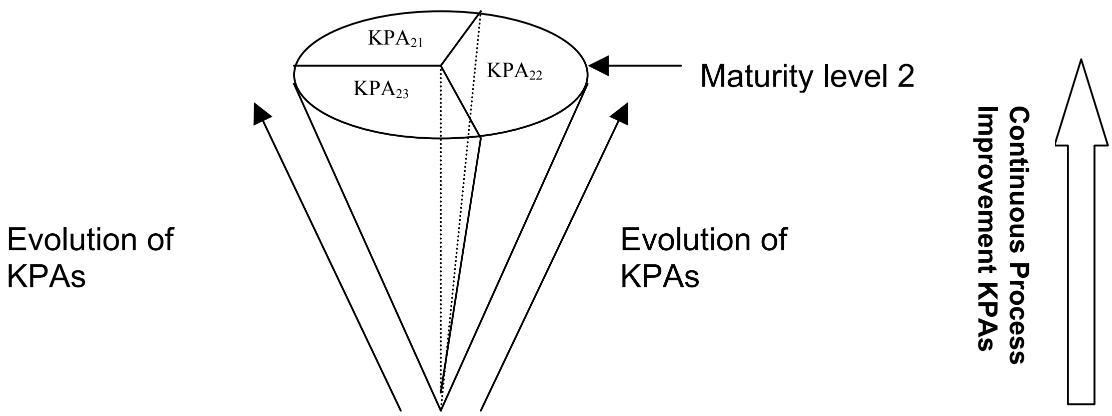


Figure 9.3: Evolutionary process of maturity level 2 KPAs

This chapter focuses on the specific KPAs for maturity level 2, while chapter 8 provided an in-depth discussion on the continuous process improvement group of KPAs (M_c).

9.3 Time management

Time management refers to the processes involved in the delivery of software projects within a specified time period. Late delivery of software is a major source of conflict between software development organisations and clients with the result that a software contract may be cancelled and penalties claimed as damages. The matter may not be as simple as that for the clients since one software project may depend on the successful delivery of another one. It is, therefore, important for software developers to be committed to fulfil their contracts as agreed upon with their clients.

Time management involves the tracking of software projects as per the initial plan. Time management is mapped onto 4 process groups as per the ISO 9001:2000 standard. The development of the time management KPA is a requirement for an organisation to achieve maturity level 2. The KPA is developed progressively until it attains maturity at level 2.

It is plausible that a good starting point for time management is to produce a Work Breakdown Structure (WBS) or Product Breakdown Structure (PBS) and generate a list of activities. These activities can each be assigned a duration for completion. It must be noted that some activities may follow others, while other activities may occur concurrently. In ESPM^3, time management can be carried out from the following inputs:

- List of activities.
- Constraints.
- Resources.
- Assumptions.
- Historical data.

In ESPM^3, the following outputs are generated:

- A project schedule.

- Documentation.

The transformation of the inputs to the outputs is shown in figure 9.4:

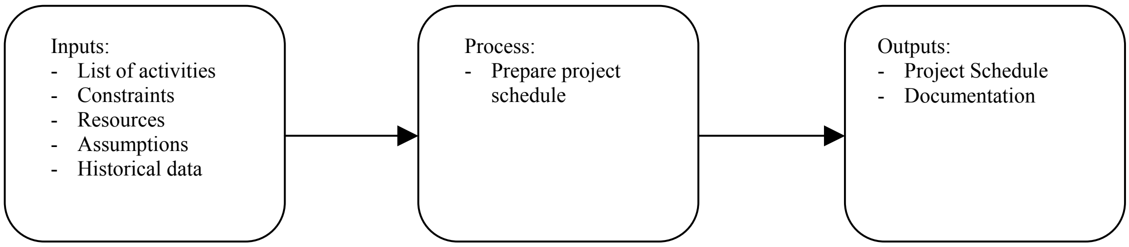


Figure 9.4: Transformation of inputs into outputs

Time management is mapped onto four process groups, namely plan, do, check and act as per figure 9.5.

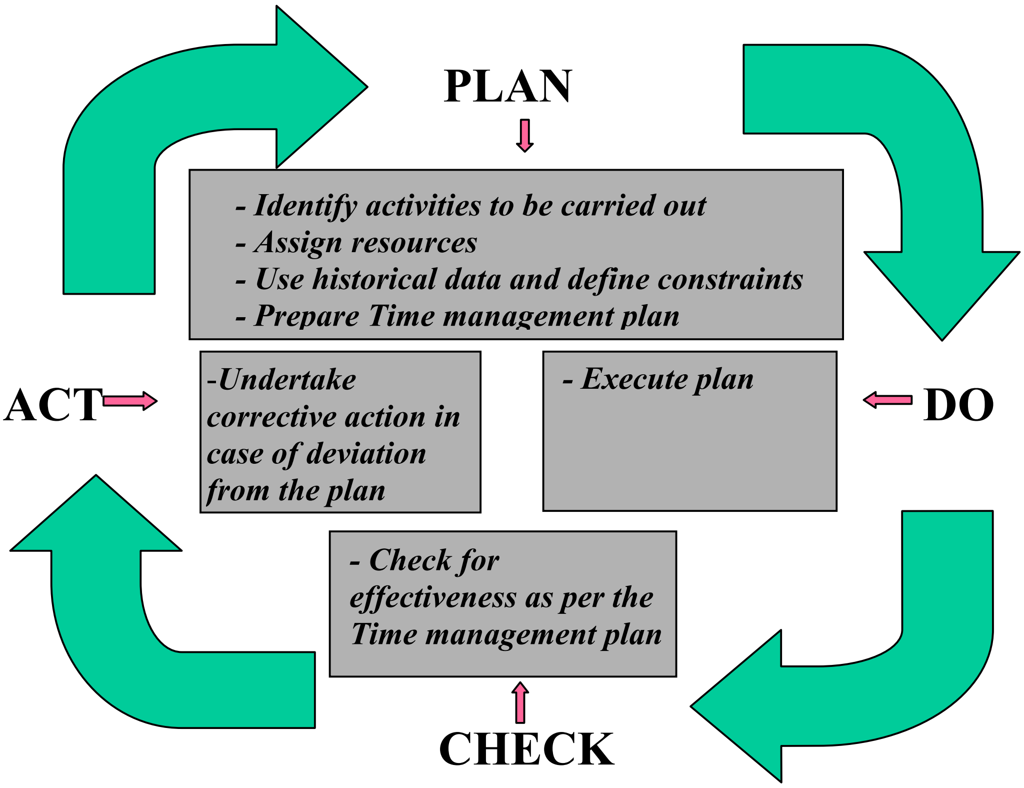


Figure 9.5: PDCA cycle for Time management

9.3.1 Plan: Time management

Time is a very important resource in project management. According to Peter Drucker (cited in Gray and Larson, 2000), time is the scarcest resource. Unless it is managed nothing else can be managed. The planning stage sets the target for the overall project as well as for the different tasks to be undertaken. The use of a time schedule renders the duration of the project and its activities visible. Planning the project's duration in $ESPM^3$ can be effected by several tools and techniques like:

- Time estimation by analogy.
- Expert judgement.
- Simulation.
- Gantt chart.
- Network diagram.

The use of Gantt charts is quite popular in Mauritius. According to a survey carried out by Sukhoo et al. (2004b), around 70% of software development organisations make use of Gantt charts to generate their project schedules. Network diagrams are not as popular in Mauritius. According to the survey carried out by Sukhoo et al. (2004b), around 20% of software development organisations made use of network diagrams to plan the progress of software projects. However, network diagrams may provide useful information and can be considered together with Gantt charts along with the proposed $ESPM^3$.

9.3.2 Do: Time management

The execution of a project is carried out according to the plan set out in the previous process group. As the activities are carried out, the progress is displayed so that the actual progress can be compared with the planned one. The Gantt chart provides useful information on the progress of activities. For example in figure 9.6, the progress of the actual analysis activity of a hypothetical project represents 50% completion (represented by the thick line in the horizontal column), while the other dependent activities have not yet started.

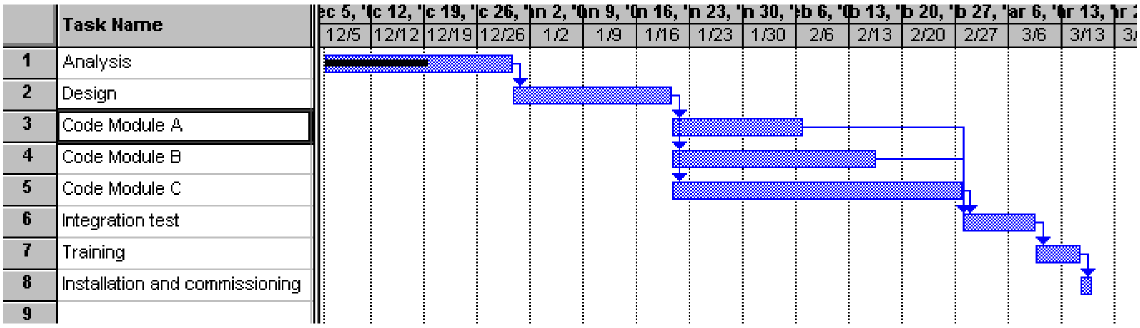


Figure 9.6: Gantt chart showing actual progress of the analysis activity for a hypothetical project

9.3.3 Check: Time management

Measurements are taken regarding the progress of the project. The actual progress is compared against the planned progress. An actual progress exceeding the time that was planned indicates a schedule overrun. This can cause delays in other activities, thus, resulting in the project running behind schedule.

However, a premature completion of an activity is welcomed as it represents a saving in cost, provided other activities are not delayed later.

9.3.4 Act: Time management

Any deviation from the plan as far as a delay in the completion of an activity is concerned is dealt with during the act process group. The reasons for the deviation need to be identified and action is taken to bring the project within schedule. Depending on the nature of the problem, the project manager may choose among a number of solutions like:

- Adding more people on the project.
- Extending the hours of work.
- Outsourcing certain activities.

However, the real-life situation is even more complex. Adding more people to a project or on certain tasks may not always reduce the time factor as discussed by Sukhoo et al. (2004a). The project manager and software team members may deal with multiple project executions at the same time. In such a case, a master schedule needs to be prepared to provide a bigger picture of all projects being managed. The master schedule prepared in the form of a Gantt chart displays the various projects undertaken as well as the progress of each project. An example is shown in figure 9.7.

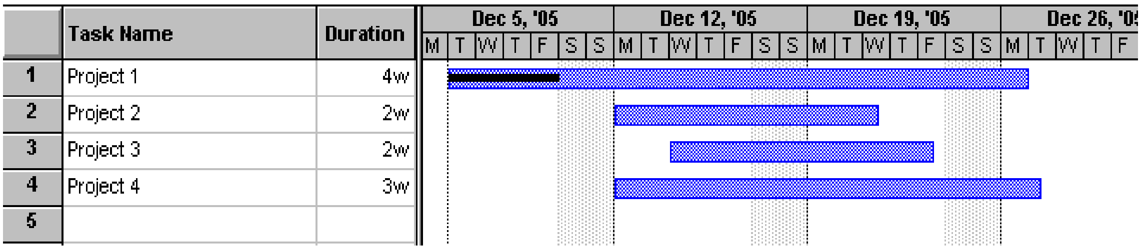


Figure 9.7: Master schedule showing progress of Project 1

9.4 Cost management

Cost management refers to the process that ensures that a software product is delivered within the agreed cost. The cost may be an integral part of any software project management plan. Underestimation of the cost of software results in the loss of money on the part of the developer while overestimation results in a loss of the contract as the client may decide not to go ahead on the basis of a cost-benefit analysis or return on investment.

According to Schach (2002), two types of cost are associated with software development. The first type is called internal cost, which is the cost to the developers while the second one is called external cost and it is the cost to the clients. The external cost is the sum of the internal cost and a profit.

Cost management can begin with the identification of all inputs and the process to transform them into outputs. A Work Breakdown Structure (WBS) or Product Breakdown Structure (PBS) is used as input along with the resources required and historical information. In ESPM[3], cost management can be carried out from the following inputs:

- Work Breakdown Structure.
- Project schedule.
- Resources.
- Historical data.

The inputs of the process generate outputs in the form of:

- Cost estimation.
- Documentation.

The transformation of the inputs to the outputs is shown in figure 9.8:

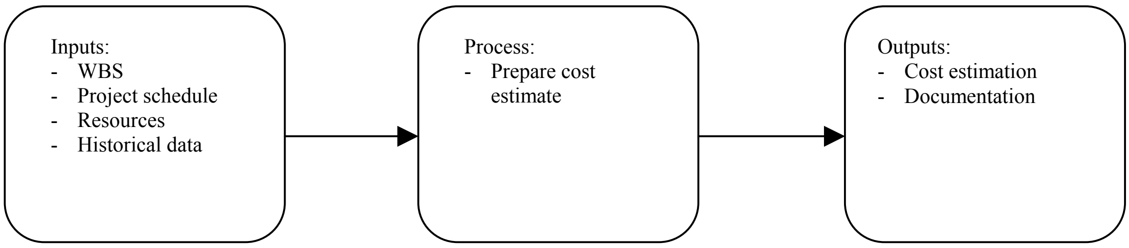


Figure 9.8: Transformation of inputs into outputs

Cost management is mapped onto four process groups (the PDCA cycle), namely plan, do, check and act as shown in figure 9.9.

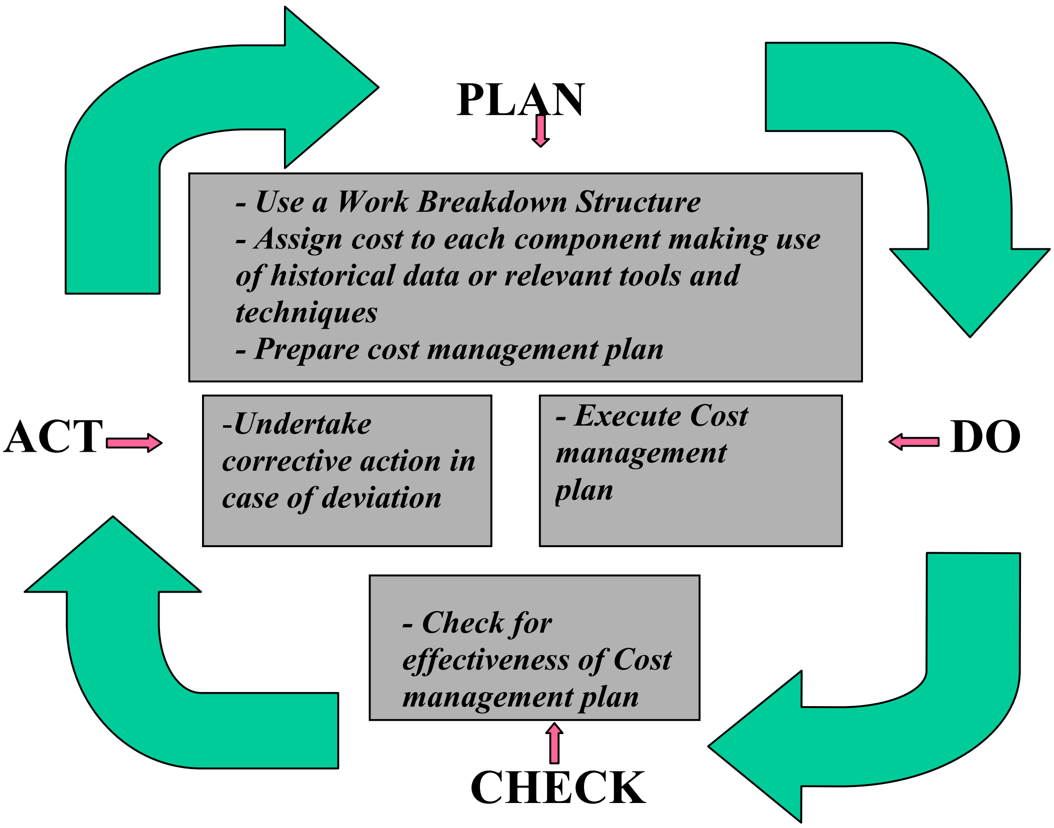


Figure 9.9 PDCA cycle for Cost management

9.4.1 Plan: Cost management

Planning for cost is fundamental to the software development organisation so as not to cause an overrun leading to the loss of money. The cost of a software product is also linked to the duration of the project. If the project is stretched over a longer period, the actual cost may deviate from the planned one. The difficulty of software estimation lies in the hidden complexity and invisibility features inherent in software. It is only when the project progresses that a real cost is available, particularly when the complexity and invisibility features unfold. In ESPM[3], planning the project's cost can be effected by several tools and techniques like:

- Analogy.

- Expert judgement.
- FFP (Files, flows and processes).
- COCOMO (Constructive Cost Model).
- Function point analysis.
- Object point analysis.

Many organisations in Mauritius are using 4GL development platforms to build applications in Oracle, Microsoft Access, SQL Server, etc. (Sukhoo et al., 2004b). Object points analysis (Hughes and Cotterell, 2006; Parthasarathy, 2008) can provide a useful means to estimate the cost of software. This technique uses the number of screens, reports and 4GL components as well as complexity (measured in terms of simple complexity, medium complexity and high complexity) weightings. As many objects may be available within an organisation and do not require development, the number of object points is adjusted. For example, if 20% of objects are already available and a total of 840 object points have been computed, the adjusted number of object points (NOP) is given by (Hughes and Cotterell, 2006):

NOP = 840 * (100-20)/100 = 672

Furthermore, the number of person-months is computed by dividing a productivity rate (PROD) into the NOP. Past data is used to determine the PROD as shown in table 9.1.

Table 9.1: Productivity rate (Hughes and Cotterell, 2006)

Developer's experience and capability	Very low	Low	Nominal	High	Very high
PROD	4	7	13	25	50

Therefore, for a development environment where productivity is nominal, the estimated effort is 672/13 = 52 person-months. The cost of development is then calculated based on the rate per person-month for the organisation.

9.4.2 Do: Cost management

As the software development project progresses against its schedule, the different internal costs involved are recorded. A breakdown of the cost per module can be kept (see figure 9.10). These costs can be useful for later comparison purposes especially when a bottom-up estimation strategy has been applied. The cost involved may be tabulated using a software tool like Microsoft Project as shown in figure 9.10.

	Task Name	Fixed Cost	Total Cost	Baseline	Variance	Actual	Remaining
1	Analysis	50.00$	50.00$	0.00$	50.00$	20.00$	30.00$
2	Design	50.00$	50.00$	0.00$	50.00$	0.00$	50.00$
3	Code Module A	20.00$	20.00$	0.00$	20.00$	0.00$	20.00$
4	Code Module B	10.00$	10.00$	0.00$	10.00$	0.00$	10.00$
5	Code Module C	25.00$	25.00$	0.00$	25.00$	0.00$	25.00$
6	Integration test	25.00$	25.00$	0.00$	25.00$	0.00$	25.00$
7	Training	30.00$	30.00$	0.00$	30.00$	0.00$	30.00$
8	Installation and commissioninc	10.00$	10.00$	0.00$	10.00$	0.00$	10.00$
9							

Figure 9.10: Sample cost details of a hypothetical software project

9.4.3 Check: Cost management

This process group allows the software developer or project manager to measure and compare the actual cost with the planned cost as the project proceeds. This is carried out for all activities to ensure that they are in line with the planned estimates.

9.4.4 Act: Cost management

Any deviation from the planned estimates must initiate an action on the part of the project manager to determine the reasons for any significant cost overrun. The use of appropriate software tools is recommended. For example, Microsoft Project or any other suitable software tool can show the progress of the activities along with their planned and actual costs. Therefore, the project manager can track cost at a glance with such software tools, provided the database is always kept up to date.

9.5 Quality management

By adapting the definition of Webster's New World Dictionary, software quality can be defined as the degree of excellence of the software. The purpose of quality management is to ensure that the project will satisfy the needs for which it was undertaken (Schwalbe, 2004). According to Munro-Faure L., Munro-Faure M. and Bones (1994), the benefits that can accrue from the implementation of a successful quality management system are:

- Improved customer satisfaction.
- Elimination of errors and waste.
- Reduced operating costs.
- Increased motivation and commitment from employees.
- Increased profitability and competitiveness.

Quality management can begin with the identification of all inputs and the processes to transform them into outputs. The following inputs to the quality management process are identified:

- User requirements.
- Quality standards.
- Functional specifications.
- Quality measurement criteria.

The processes generate outputs in the form of:

- Quality management plan.
- Quality management checklists.
- Products conforming to standards and criteria.

The transformation of the inputs to the outputs is shown in figure 9.11:

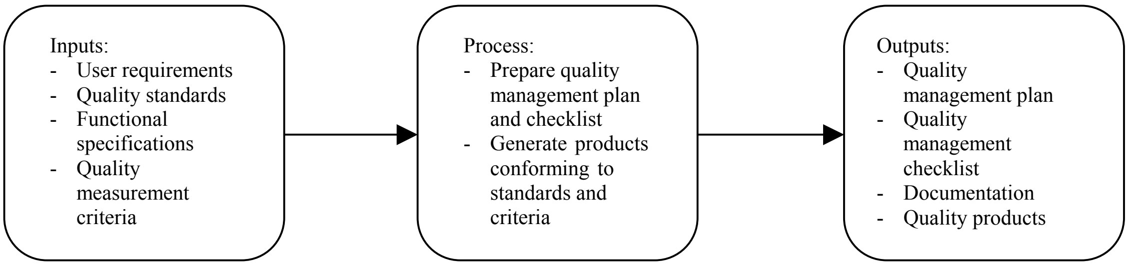


Figure 9.11: Transformation of inputs into outputs

Quality management is mapped onto four process groups (the PDCA cycle), namely plan, do, check and act as per figure 9.12.

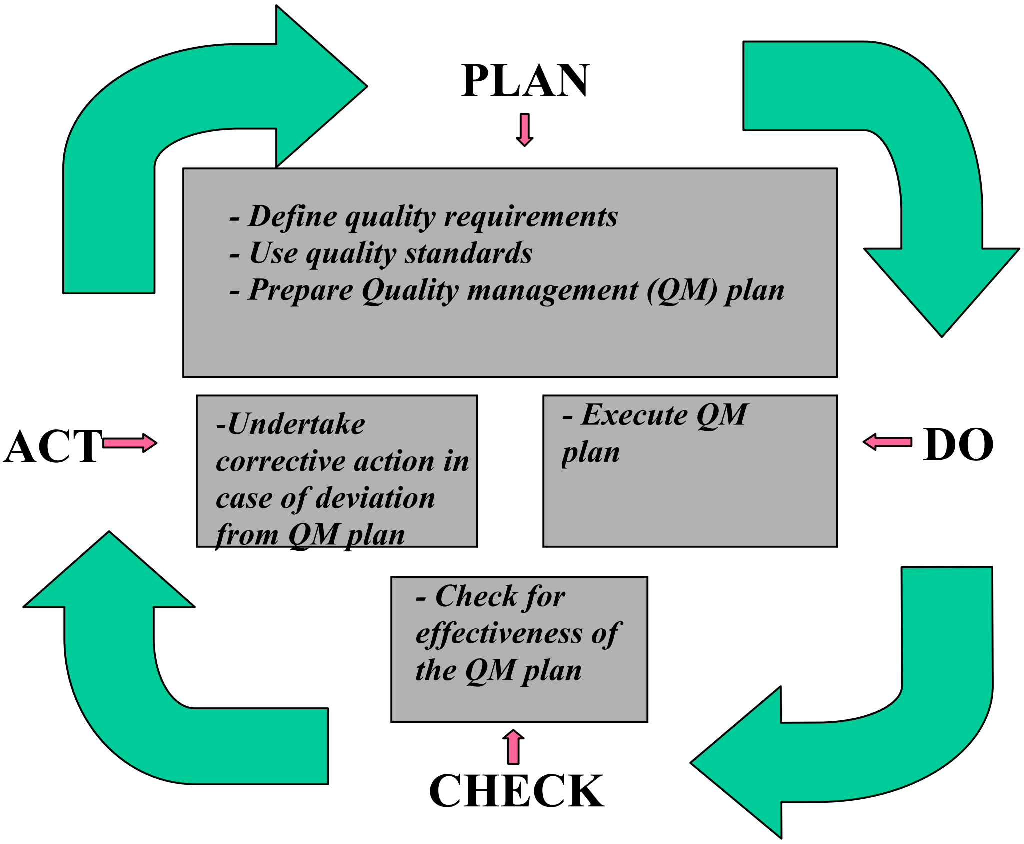


Figure 9.12: PDCA cycle for Quality management

In ESPM[3], quality management is initiated by defining the following criteria:

- Functionality.

- Usability.
- Reliability.
- Performance.
- Serviceability.

The implementation of a quality management system by the developer organisation may add other criteria to the above list.

9.5.1 Plan: Quality management

In ESPM[3], a quality plan is prepared prior to the development of any software product. The desired quality targets need to be specified at this stage. The approach to be used, the implementation strategy and the major products have to be made clear at this stage. In the case of Mauritian software development organisations, ISO 9001:2000 certification is increasingly being sought after. Therefore, a quality management system to be implemented in the Mauritian context needs to focus on the requirements of ISO 9001:2000.

9.5.2 Do: Quality management

The plan is executed during the development of the software and associated documentation. The implementation of the plan has to be closely monitored. The use of checkpoints to determine quality at defined stages can provide useful measures.

9.5.3 Check: Quality management

The verification of the level of quality relies upon the criteria set out in the plan. During the “Do” process group, the data gathered at various checkpoints is verified against the predefined criteria to find out about any deviation.

9.5.4 Act: Quality management

This process group is concerned with the necessary action in case of deviation in software quality. It ensures that the product conforms to the initially agreed upon quality level. This may also involve a change in plan and revisiting of the other process groups.

9.6 The evolutionary process from maturity level 1 to maturity level 2

The evolution to higher maturity levels depends on the organisation's commitment and availability of resources, experience as well as skills. For some organisations, this process may take longer than for others. Therefore, a yardstick is required to enable an organisation to measure its progress.

The maturity of the KPAs is determined through an assessment method that is explained in section 11.7 of chapter 11. As shown in figure 9.3, the KPAs grow from an empty set (at maturity level 1) to a union of three fully mature sets at level 2. An organisation, therefore, has to build its three KPAs through a sequence of iterations. Ensuring that projects are completed within time, cost and quality constraints is the motto of this evolution process.

9.7 Conclusion

In this chapter, the maturity level 2 KPAs were discussed in greater detail. Maturity level 2 is considered as the basic project management level whereby projects are managed so as to improve delivery on time, within budget and according to the required quality standard.

Time management KPA focuses on the time factor for the development of software. Progress in the development of a software product is not directly visible as software is different from traditional engineering products as explained in section 8.5. The progress in the construction of a building is visible. Time management is made possible after the invisible features are directly rendered visible through certain tools and techniques like the use of

Microsoft Project software tool and techniques like Gantt charts and the Critical Path Method. The PDCA cycle takes the time management KPA through a sequence of stages to ensure that ISO requirements are adhered to.

Cost management in ESPM^3 aims at ensuring that the cost of a software product reflects the amount of effort required in its development and may allow an organisation to make a reasonable profit. Again, this requires rendering the effort needed visible. Breaking down of large and complex software into smaller and simpler components and assigning a monetary value to each component based on the amount of effort required for its development is the most common method used. The cost management KPA focuses on the management of the cost parameter so that the risk of a cost overrun is reduced.

Lack of quality is of great concern to users who are more conscious of such requirements than in early days of software development. User requirements, performance, security features, user friendliness of an interface are all quality issues that are of concern. The quality management KPA focuses on all the quality features that are determined during the planning process group of the PDCA cycle and aims at addressing them during the software development process.

The basic project management level (maturity level 2) incorporates the basic KPAs for software project management. The ESPM^3 may contribute to the delivery of software products on time, within budget and as per quality standards through its three basic KPAs.

In the following chapter, KPAs of maturity level 3 are discussed. They operate at the organisational level, thus ensuring a consistent approach across the organisation. In large organisations, specialised departments may be responsible for addressing these KPAs so that project managers can focus on core activities.

Chapter 10

Maturity Level 3

Table of contents

10.0 Chapter layout

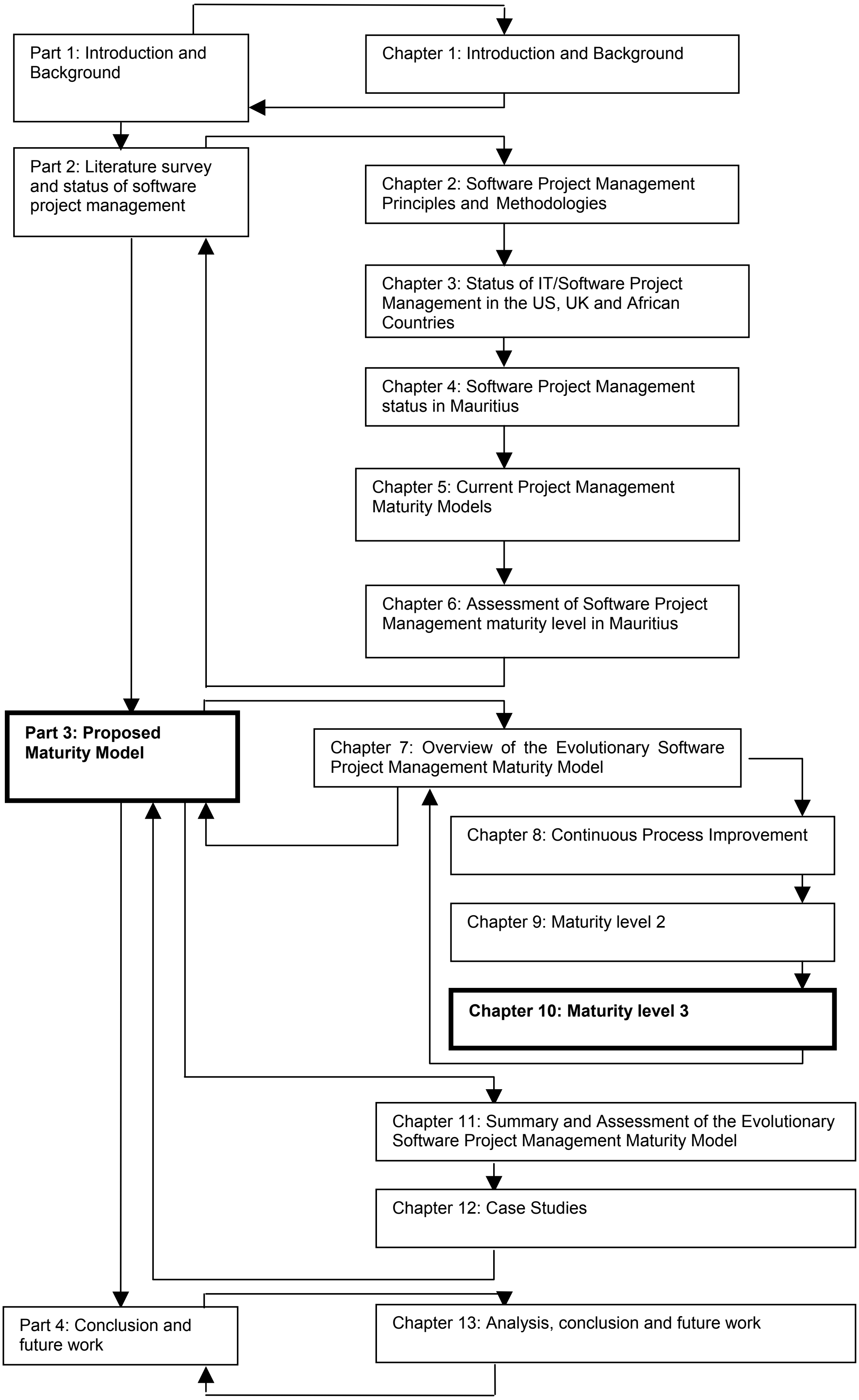


Figure 10.1: Chapter 10 in context within the overall thesis

10.1 Introduction

In chapter 9, Maturity level 2 was discussed as the basic project management level with KPAs to improve the success rate of projects in terms of delivery of software with time, cost and quality constraints. In addition, the KPAs of the continuous process improvement group provided a facilitating function.

In the current chapter, maturity level 3 KPAs are discussed. Maturity level 3 of the ESPM^3 is also called the organisational level as KPAs at this level are applied consistently across the organisation. Maturity level 3 comprises three KPAs, namely:

- Human resource management.
- Risk management.
- Contract management.

The above KPAs support the level 2 KPAs by ensuring a facilitating function. For example, human resources are chosen carefully in order to provide an appropriate blend of complementary skills for software development to be carried out within time, budget and quality standard. Risk management is sometimes a specialised function executed by specialists, otherwise project managers are given the responsibility to perform risk management according to agreed guidelines within the organisation. Contract management is handled according to standard contract document prepared for each organisation.

Contract management may not be relevant to organisations that develop software for in-house use. However, the KPA is discussed for organisations where contract management is applicable.

In addition to the discussion on the three key process areas at maturity level 3, emphasis is placed on tools and techniques (such as responsibility assignment matrix and risk analysis table) available to ensure that software projects are well managed and risks are mitigated or eliminated. Suggestions are also made about the tools and techniques appropriate for Mauritian software development organisations.

10.1.1 Aim, objectives and structure of this chapter

In chapter 10, the maturity level 3 KPAs are described in greater detail than in chapter 7. The application of the Plan-Do-Check-Act cycle to the various KPAs is also explained.

The remainder of the chapter is structured as follows:

Section 10.2 – In this section, a conceptual representation of maturity level 3 is shown diagrammatically and explained.

Section 10.3 to 10.5 – All the KPAs are described in detail in these sections and the PDCA cycle as applied to these KPAs is made clear.

Section 10.6 – In this section, the evolution from maturity level 2 to level 3 is explained.

Section 10.7 – In this section, it is concluded that the KPAs of maturity level 3 have an organisation-wide focus. Any KPA that is not relevant to an organisation is ignored and its assessment is not required to evaluate the overall maturity level of the organisation.

10.2 KPAs at Maturity level 3

Maturity level 3, M_3, is the union of all KPAs at level 3 and level 2 (M_2). M_2 includes the continuous process improvement group of KPAs (M_c). Conceptually, maturity level 3 KPAs are found around maturity level 2 KPAs as shown in figure 10.2.

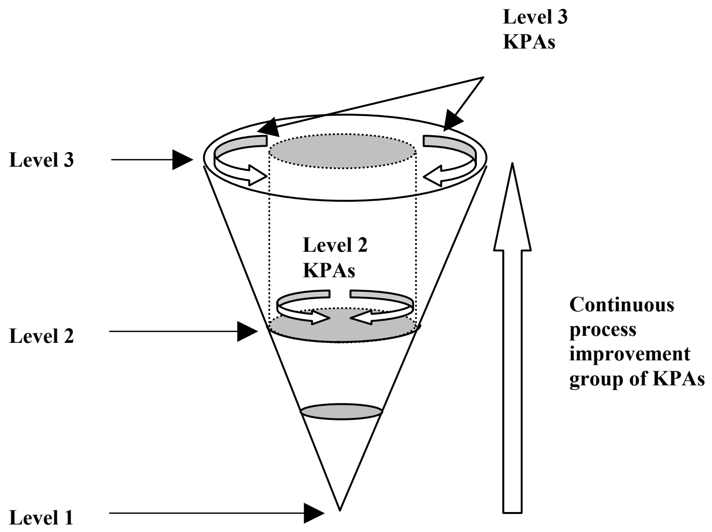


Figure 10.2: Maturity level 3

The layout of maturity level 3 KPAs around level 2 KPAs was shown in chapter 7 and is reproduced in figure 10.3.

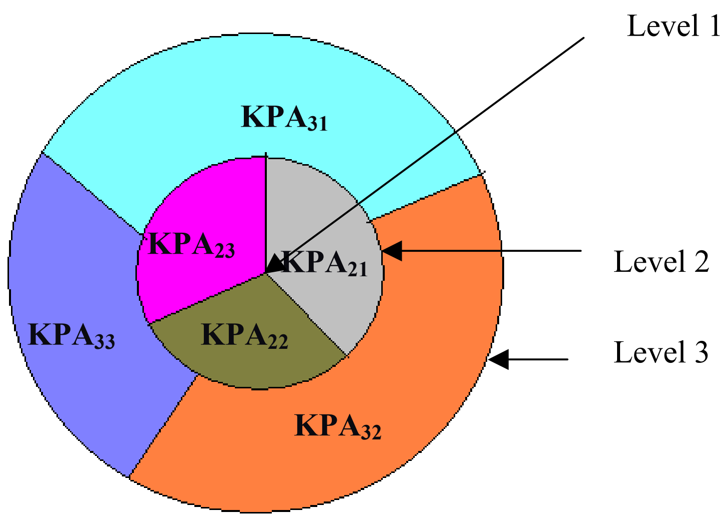


Figure 10.3: Layout of maturity level 3 KPAs

M_3 is given by

$$M_3 = \bigcup_{i=1}^{m} KPA_{3i} \cup M_2 \qquad (10.1)$$

where m is the total number of KPAs at level 3 (the KPA_3's) and M_2 is given by (10.2). This means that M_3 comprises KPAs specific to level 3 (human resource management, risk management and contract management KPAs) and the KPAs of level 2.

$$M_2 = \bigcup_{j=1}^{3} KPA_{2j} \cup M_c \qquad (10.2)$$

Maturity level 2 in turn comprises those KPAs specific to level 2 (time management, cost management and quality management) as well as those KPAs of the continuous process improvement group. Maturity level 1 has no KPAs (the empty set, Ø) and, therefore, does not contribute towards the overall list of KPAs.

Therefore, at maturity level 3, all KPAs at maturity level 2, the continuous process improvement group of KPAs as well as the specific KPAs for maturity level 3 are applicable. Therefore, maturity level 3 provides a comprehensive software project management framework for software development. Maturity level 3 provides the highest level of sophistication for software development and continuous process improvement ensures that the organisation continuously adapts to changes that occur within its environment.

10.3 Human Resource Management

Human resource is considered as the key resource for competitiveness in an organisation. According to Nel, Gerber, Van Dyk, Haasbroek, Sono and Werner (2001:4), it is viewed as "[t]he only resource in an organisation that reacts when acted upon.". In order to maximise the output of human resources, it is critical to manage these resources efficiently. Human Resource Management (HRM) refers to the policies, practices and systems that influence employees' behaviour, attitudes and performance (Noe, Hollenbeck, Gerhart and Wright, 2004).

HRM encompasses a range of practices that collectively aims at influencing an organisation's performance (Noe et al., 2004). At the project level, these practices are:

- HR planning.
- Selection.
- Employee relations.

However, today, applying HRM in the generic sense is not suitable to meet the goals of an organisation. According to Chaturvedi (2005), it has to be linked to the business strategy, hence the importance of Strategic Human Resource Management (SHRM). Friedman and Strickler (cited in Noe et al, 2003:55) defined SHRM as "[t]he pattern of planned human resource deployments and activities intended to enable an organisation to achieve its goals.".

In ESPM[3], Human Resource Management can be carried out from the following inputs:

- Project team attributes.
- Organisation structure and constraint.
- Employee relations constraint.

The processes generate outputs in the form of:

- An HRM plan.
- Documentation.

The transformation of the inputs to the outputs is shown in figure 10.4:

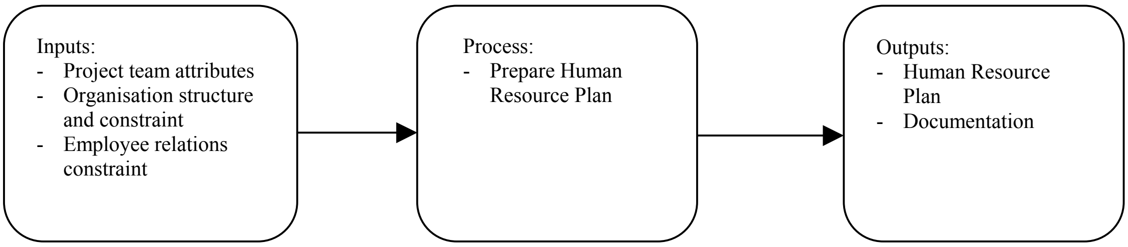


Figure 10.4: Transformation of inputs into outputs

Human Resource Management is mapped onto four process groups, namely plan, do, check and act as shown in figure 10.5.

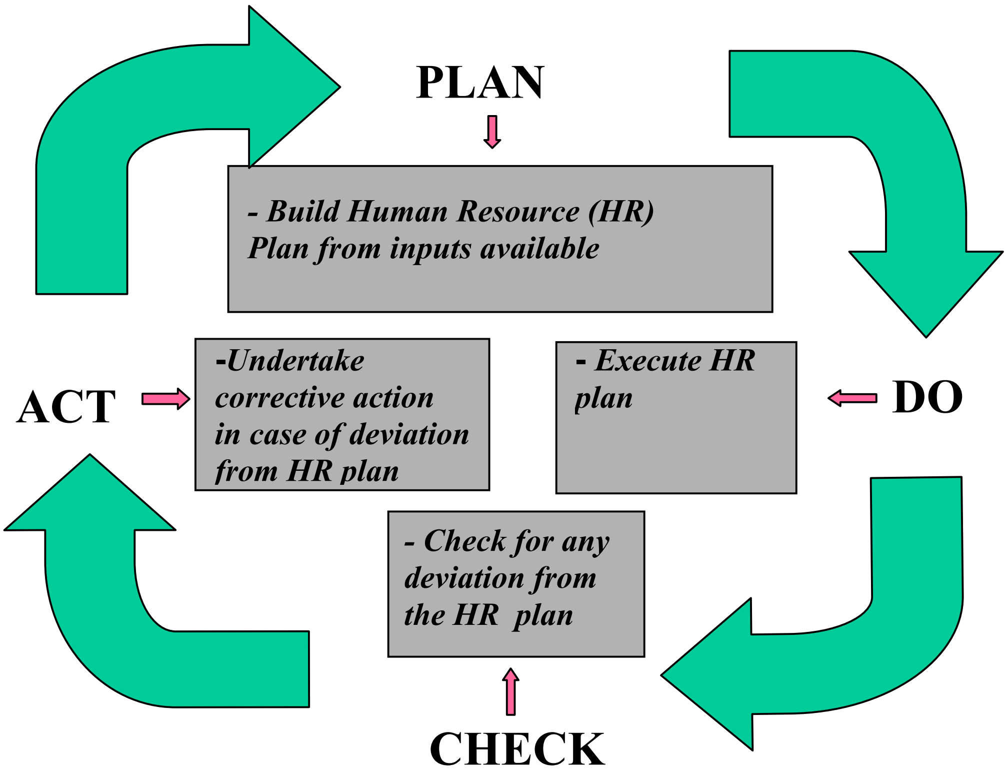


Figure 10.5: PDCA cycle for Human Resource management

10.3.1 Plan: Human Resource Management

Human Resource planning determines the numbers and types of employees to be assigned to a software project. It is an important activity since it provides for:

- More effective and efficient use of human resources.
- Greater employee satisfaction.
- Reduced conflicts at work.

One useful tool that may be considered during the planning phase is the responsibility assignment matrix. This is a table (as per table 10.1) listing the staff required based on their skills against the phases of the software development process.

Table 10.1: Responsibility assignment matrix

Staff / Phases	A	B	C	D	E	F
Feasibility	X					
Analysis		X				
Design		X	X			
Coding			X	X		
Testing					X	
Training						X
Implementation	X	X				

The planning process group also takes care of the planning for the selection of project team members and ensures a suitable employee relationship.

10.3.2 Do: Human Resource Management

This process group involves the selection of appropriate team members and consequently, the maintenance of a conducive environment while the project is in progress.

Selection is an important activity in HR planning. This activity comprises the constitution of the project team based on the criteria that are best met by individual team members to carry out the different project tasks.

Some factors that influence the selection process to be considered are:

- Type and size of project.
- Deadline to complete the project.
- Skills of available staff.

Employee relations refer to the relationships between employers, employees and the state (tripartite relationship) engaging in economic activity in an industrialised society (Swanepoel, Slabbert, Erasmus and Brink, 1999). Employment relationship is a relationship between the employer, employee and the state. Sometimes, the state may also be the employer. The state normally serves to regulate employment through legislations.

In Mauritius, it is very common to find staff members of large software development organisations represented by Trade Unions. Unlike large organisations, smaller organisations rarely comprise staff members being represented by Trade Unions. In such cases, any conflicts that cannot be solved at the level of the organisations are forwarded to the State (represented by the Ministry of Labour and Industrial Relations) or taken directly to the labour court.

A tripartite relationship normally exists to settle employee relationship conflicts. This relationship is represented in figure 10.6.

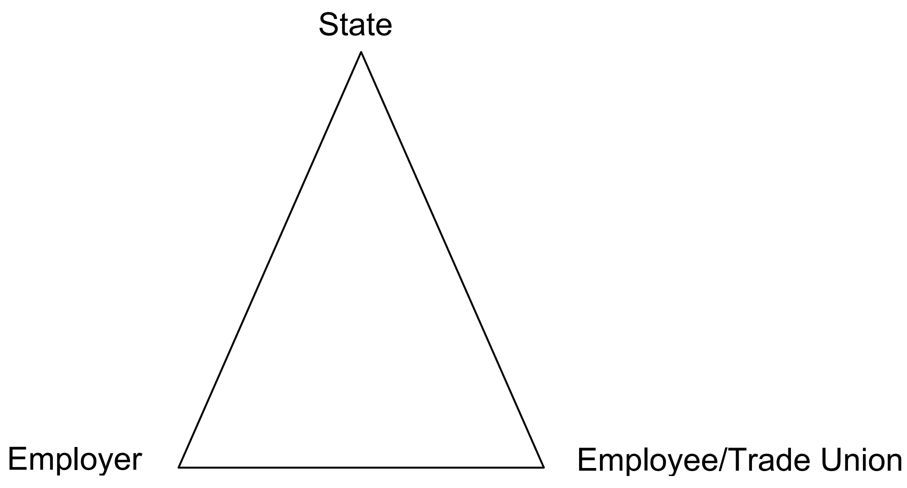


Figure 10.6: Tripartite relationship (Swanepoel, Slabbert, Erasmus and Brink, 1999)

10.3.3 Check: Human Resource Management

During this phase, the performance of the team members is monitored and conflicts are regularly identified and eliminated so as not to affect the smooth execution of the project. Some team members are taken on board for the project at appropriate phases of the software development exercise while others are removed and assigned to other projects after they have completed their tasks.

10.3.4 Act: Human Resource Management

Appropriate action is taken in the case of any deviation from the plan. The project manager may decide upon:

- Replacement of team member(s).
- Resolving conflicts using his/her acquired experience and skills.

The replacement of staff is done with due care in order not to disturb the work in progress. For example, replacement of an experienced programmer with one possessing inadequate experience can disturb the required blend of skills on a project and consequently affect the timely delivery of a software product within budget and quality constraints.

Similarly, conflict resolution must not give the impression that one party is being favoured as compared to the other party. The project manager has to adopt an impartial and objective attitude towards the parties in conflict.

10.4 Risk Management

Risk management is concerned with the mitigation or elimination of unforeseen events that may pose a threat to the successful completion of a project. It is achieved by avoiding these risks or drawing up risk management plans for dealing with them (Hughes and Cotterell, 2006). Software development is characterised by various pitfalls that need to be anticipated. A project manager is particularly concerned with risks that can cause the late delivery of a project, a budget overrun or a poor quality software product to be developed. In large and complex projects, a risk manager may be appointed to deal with risks.

In ESPM[3], the inputs are identified and the process to transform them into outputs is developed. The following inputs are considered:

- List of potential risks.
- Historical data.

The following outputs are generated:

- Prioritised risks according to their likelihood and impact.
- Risk management plan.
- Documentation.

The transformation of the inputs to the outputs is shown in figure 10.7:

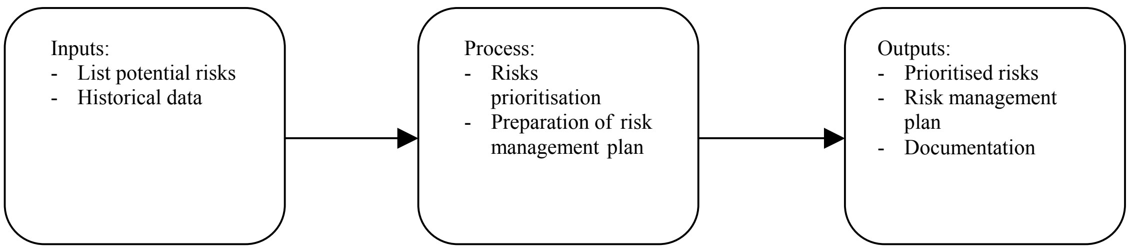


Figure 10.7: Transformation of inputs into outputs

Risk management is mapped onto four process groups, namely plan, do, check and act as indicated in figure 10.8.

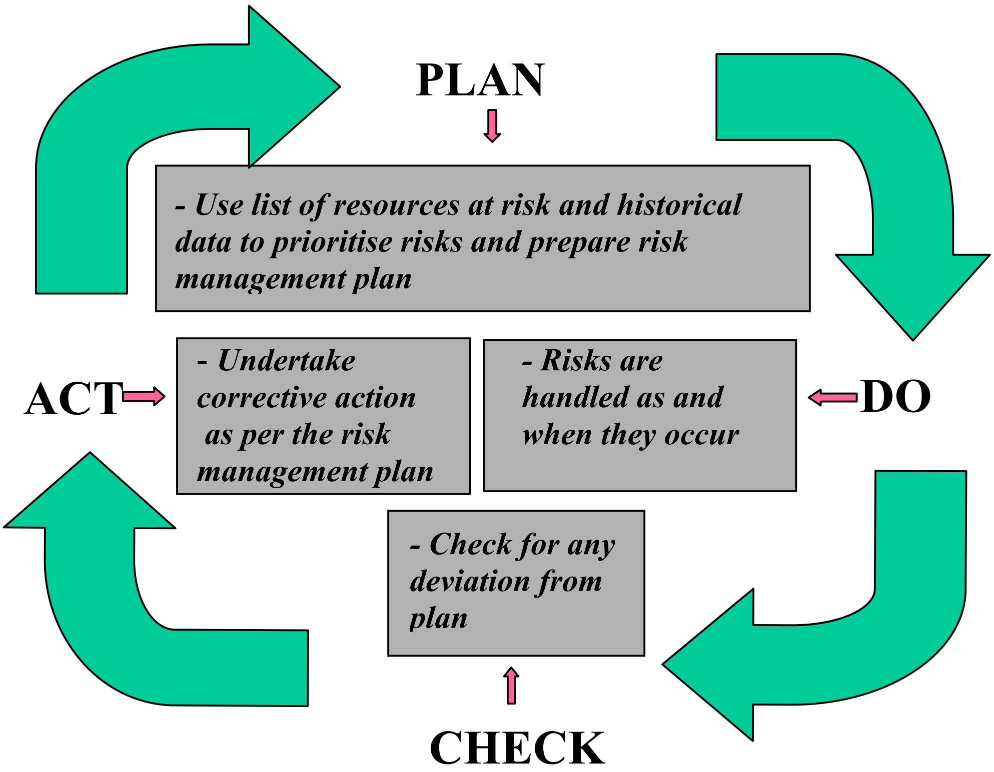


Figure 10.8: PDCA cycle for Risk management

10.4.1 Plan: Risk Management

Risk planning consists of drawing up risk management plans and, where appropriate, adding them to the project's task structure (Hughes and Cotterell, 2006). In the ESPM[3], this process group is also concerned with the identification of those risks that may pose a threat to the project, their evaluation and prioritisation.

Risk identification refers to the potential hazards that may affect the project success. During the planning phase of the project, all such potential hazards affecting the project resources are listed.

Risk evaluation is concerned with the determination of the severity of impact that a particular risk may pose. A simple method that can be implemented is to assign values between 1 and 5 on a Likert scale to the risks identified. These values can have the following denotations:

1 = No impact
2 = Low impact
3 = Moderate impact
4 = High impact
5 = Very high impact

A project manager is also free to assign values from 1 to 10 for the severity of the impact. Similarly, the likelihood that an identified risk hazard occurs is also assigned a value from 1 to 5 or from 1 to 10 depending on the project manager.

An example of a risk analysis table (Hughes and Cotterell, 2006) that may be prepared is shown in table 10.2.

Table 10.2: Risk Analysis Table

Risk	**Likelihood**	**Impact**	**Risk exposure**
Changes to specifications	1	4	4
A key staff member being ill during critical activities	1	3	3
Coding taking more time than planned	2	4	8
Users not satisfied with the quality of the software in terms of functionality	1	5	5

Risk exposure is the product of the likelihood of a risk hazard occurring and its impact on the successful delivery of the project. For example for the first row in the table 10.2, the Risk exposure is calculated as:

Risk exposure = Likelihood * Impact

= 1 * 4

= 4

The risks are then prioritised in terms of their risk exposure value. A new table, sorted in descending values of risk exposure, can then be generated. This new table will allow project managers and developers to focus on high risks associated with project tasks concerned.

A risk management plan, elaborating the means to deal with a risk hazard, is also drawn up during the planning phase. This plan allows the project manager to adopt a proactive measure to deal with risks instead of adopting crisis management.

10.4.2 Do: Risk management

Risk management is carried out using the risk management plan developed in the "Plan" process group. The risk management plan elaborates on the risks involved and the means to eliminate or mitigate them if they occur. During the execution stage of a project, the risk management plan is also executed and potential risks at appropriate instances are addressed proactively.

A change in legislation that may cause a change in the calculation of income tax can have a high impact on a payroll project. This may lead to a budget overrun and a schedule overrun. Such a project has to take into consideration the impact and make provision for an increase in budget and an extension in the delivery date. The contract between the software development company and the client must also make provision for such a risk. Therefore, all risks identified need to be addressed proactively should such an event occur.

10.4.3 Check: Risk management

This process group allows the software developer or project manager to monitor any risks that were identified during the planning process group. Risk monitoring must be an ongoing activity as the likelihood or impact of a risk may change as the project proceeds (Hughes and Cotterell, 2006). For instance, the illness of a staff member during the specifications phase may have a low impact as compared to during user training.

In this process group, the effectiveness of dealing with a risk hazard is checked. For instance, the illness of a member of staff during the training phase may be dealt with by making available an alternate member of staff for the exercise. Given the importance of training to be carried out in time for the successful implementation of the project, the software development company will have to cater for such a situation by ensuring that a staff is replaced without major disturbance in the event of absence due to illness or other reasons.

Once staff replacement has been effected, the effectiveness of the training has to be evaluated so that there is minimum complaint on the part of the user. Therefore, it is important to select the replacement staff carefully by ensuring that the person is conversant with the software to be used by the users. The project manager should be able to create a conducive environment for the training to be conducted and whenever required has to intervene to address any issue that may initiate a complaint.

Should a risk hazard occur, the next process group is considered.

10.4.4 Act: Risk management

Any risk hazard requires rapid intervention on the part of the project manager to bring the software development process within control. Any deviation in the expected outcome of the project is dealt with as far as possible.

For example, the risk of a key staff leaving the project while it is in a critical stage of development may severely impact on the success of the project. The project manager must be aware of such a possibility and appropriate action needs to be taken. At the organisational level, measures to counteract such a risk must be properly documented at a general level. For a specific project, the project manager must have a plan of action, prepared during the planning phase, to address such a risk. In this context, a properly designed risk management plan comprising a list of comprehensive risks can address risks hazards proactively.

However, any action to mitigate or eliminate risk hazards needs to be properly documented in the risk management plan. In this way, crisis management may be avoided.

10.5 Contract management

A contract is a formal agreement between two parties wherein one party (the contractor) obligates itself to perform a service and the other party (the client) obligates itself to do something in return, usually in the form of a payment to the contractor (Gray and Larson, 2000). The supply of a software product by a software development company to a client is contractual in nature. In Mauritius, a fixed-price contract is favoured for the supply of a software product. Under such a contract, the contractor agrees to perform all work specified in the contract at a fixed price. An invitation for bids is requested by the client and the final choice can depend on the evaluation criteria used. The tendering process can be one of the following (Public Procurement Office, 2008):

- Open tendering process where any supplier can supply software products requested. All bids that are compliant to the specifications spelt out in the tender document are evaluated in the same way. The process may be time consuming, especially when there is a large number of bids.

- Restricted tendering process where bids are invited only from a restricted number of suppliers. This is done when the client does not want to spend time in evaluation by resorting to an open tendering process.
- Negotiated procedure where negotiation is initiated with one or more suppliers and the best option in the interest of the client is chosen. This procedure may be carried out when there is a time constraint, for example, in the case where a fire has destroyed part of an office including its IT equipment and the computer system has to be up and running as soon as possible.

  Similarly, the development of a module as an extension of the existing system may be subject to a negotiated procedure as the supplier is already aware of the system in use. In order to avoid any conflict between the original system and the new module and to facilitate integration of the two, it may be safer not to invite bids from other suppliers.

During the survey carried out for the identification of KPAs for the different maturity levels, there were organisations not concerned with contract management as they develop software in-house for their own use as well as for several other departments without charging any fee. In such a situation, the software development company has the flexibility to ignore the contract management KPA in its ESPM[3].

Following a tender exercise where several bids are received, the most cost effective responsive offer may be selected or a combination of price and other factors may be considered. In the latter case, marks may be awarded to performance factors, experience of the company, customer service and experience of staff. A combination of technical and financial weightings may be used to determine the best offer.

The best option, based on the combination of the technical and financial weightings, is selected and an award is made to the prospective bidder. The acceptance of an offer by the bidder is followed by the signature of a contract (sometimes negotiation may also precede this event). As from the signature of the contract by both the client and the contractor, contract management becomes an important area of focus during the development of the software.

In ESPM[3], contract management makes use of the following inputs:

- Specifications document prepared by client.
- Tender proposal from the contractor.
- Clauses in the contract and deliverables.

The output generated is:

- Successful contract execution.
- Acceptance of deliverables by the client.

The transformation of the inputs to the outputs is shown in figure 10.9:

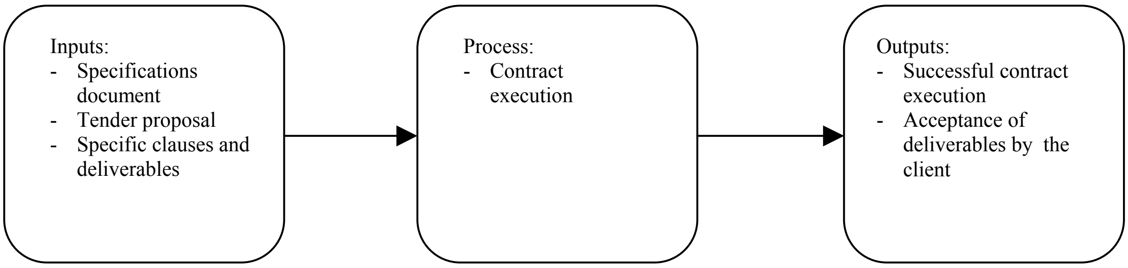


Figure 10.9: Transformation of inputs into outputs

The project team needs to be aware of the implications of the clauses in the contract. Quite often, team members seem to ignore the action that may be taken by the client with respect to the development of the software not conforming to initially agreed conditions. By default, the contract is subject to remedy in a court of law.

The mapping of contract management onto the four process groups, namely plan, do, check and act is shown in figure 10.10.

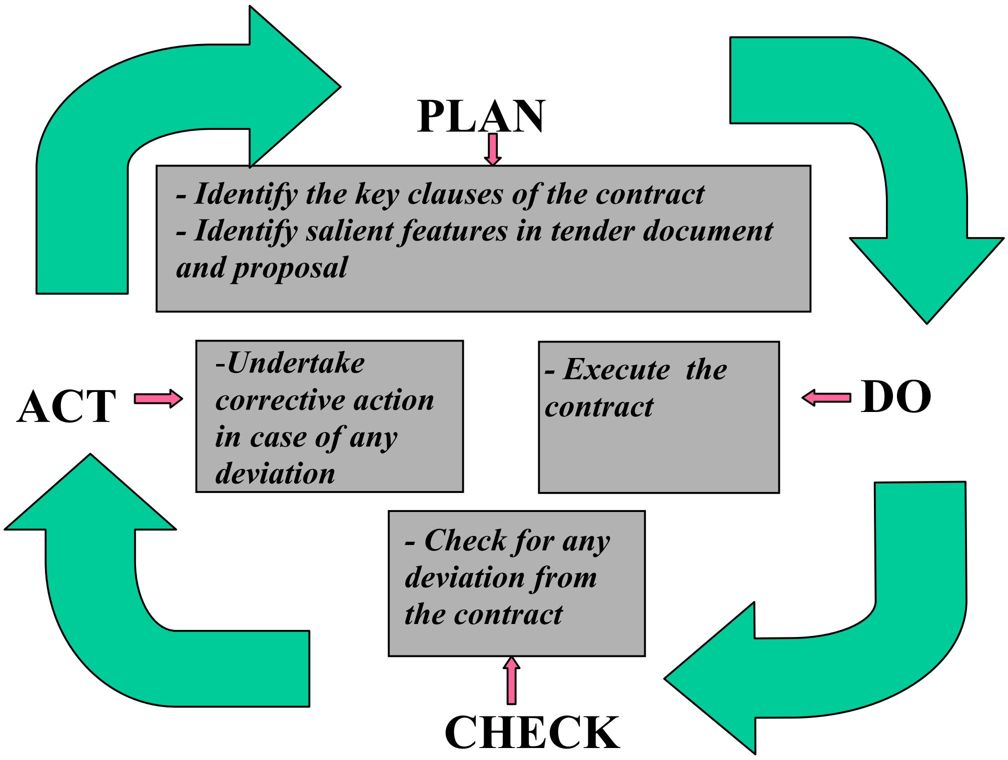


Figure 10.10: PDCA cycle for Contract management

10.5.1 Plan: Contract management

The software development company has to be proactive with regard to the clauses spelt out in the contract so as not to be liable to penalties. The tender document as well as the supplier's bid are normally binding. Project managers have to follow up closely all deliverables that need to be supplied at specified points in time. A table of deliverables and the corresponding date as per the contract may be prepared.

Follow-up is made on the clauses of contract as well as pertinent issues in the tender document and the supplier's proposal. An example of issues that require strict follow-up and actions are the payment terms. The client must observe strict compliance to payment terms as agreed on before. For instance, payment amounting to a percentage of the total contract value may only be released after the submission of a valid (in terms of the period that it has to cover) bank guarantee by the client.

10.5.2 Do: Contract management

The execution of contract management occurs during the project execution. All deliverables are noted together with their dates against the plan.

The list of pertinent issues in the contract as well as in the tender document and the supplier's proposal facilitates the project manager to ensure that all clauses forming part of the contract are strictly being adhered to. In addition, the software development organisation may advise the client in advance of any problems that may arise. For example, in case the client organisation is not ready for the installation of the software, the software development organisation may request testing, training and acceptance test to be carried out at its own premises so that payment is effected within a specified period.

10.5.3 Check: Contract management

Regularly, checks have to be made to ensure that the deliverables are according to the contract. A proactive approach needs to be adopted since any failure to abide by the contract clauses may result in undesirable consequences. For example, a contract may clearly indicate that any late delivery of the final product may require the software development company to pay damages to the client. Similarly, late payment on the part of the client may involve damages to be paid by the client.

10.5.4 Act: Contract management

In the case of deviation from the plan, prompt action needs to be initiated in order not to cause any breach of contract. Sometimes mutual arrangements between the client and the supplier may mitigate the effect of such damages.

However, in other projects, strict compliance may be observed. Late delivery of software may result in serious outcomes for any politically driven project. In such a case, the client may have to apply penalties in the case of late delivery of software. Late delivery of a software product that has major social impact has to be strictly avoided. For instance, changes to software for the payment of pension may have serious consequences. The persons who have to assume responsibilities for such problems need to be identifiedand necessary action has to be taken by parties concerned. In case approval is given late by the client for the software developer to begin with the modification of a particular piece of software, this may involve an increase in cost. In such a situation, the client may have to incur additional cost towards the software development organisation so that additional staff may be assigned to the project.

10.6 The evolutionary process from maturity level 2 to maturity level 3

After an organisation has reached maturity level 2, progress towards the higher level may be pursued in order to benefit from higher success rate of projects. Therefore, the organisation may pursue the goal of attaining maturity level 3, which is the highest level in $ESPM^3$. The time taken to reach level 3 (from level 2) depends on the organisation's commitment, availability of resources, experience as well as skills.

The maturity level of each KPA within the PDCA process group is determined with respect to the definition of the maturity levels provided (see section 11.7 of chapter 11). Details of the assessment of maturity level are provided in section 11.7.

As shown in figure 10.2, the maturity level 3 KPAs grow from maturity level 2 as soon as the organisation is committed towards achieving the next higher level. The level 3 KPAs are built through a series of iterations. An assessment method can determine whether maturity level 3 has been reached after the organisation has made progress in the success rate of software projects.

10.7 Conclusion

In this chapter, maturity level 3 KPAs were discussed. Maturity level 3 is also called the organisational level as the focus areas are handled mostly at the organisational level. For instance, the recruitment and training of staff are handled by top management or by a specialised department. Contract management is of a major concern to top management. In case contracts are not properly handled, the software development company may end up paying large sums of money to the client. Risks are also too important to be the concern of project managers only. There may be a specialised department with risk managers dealing with risks although for smaller organisations and projects, the project manager may be responsible for handling risks.

Software development companies responsible for the development of software in-house may not be concerned with contract management. In such a case, this KPA is ignored in the software project management model (ESPM^3).

The organisational level (maturity level 3) incorporates the KPAs that are dealt with by the organisation, although the project manager may in certain cases be given the responsibility to handle the issues. Measures for human resource management, risk management and contract management are adopted across an organisation so as to facilitate software project management (to achieve timely delivery of software, within budget and quality standard).

The next chapter provides a summary of the ESPM^3 in tabular forms. The tables show at a glance the specific KPAs involved, their goals as well as the relevant tools and techniques that are applicable to organisations in Mauritius. The synthesised information provides the project manager and developers with a quick reference guide to the salient elements of the ESPM^3. Instead of going through the whole documentation on ESPM^3, readers may have a quick overview of the model. In addition, project managers and software developers may use the summary as a quick reference guide to manage their projects. Project managers and software developers may, therefore, be relieved from having to browse through the whole documentation on ESPM^3.

Chapter 11

Summary and Assessment of ESPM^3

Table of contents

11.0 Chapter layout

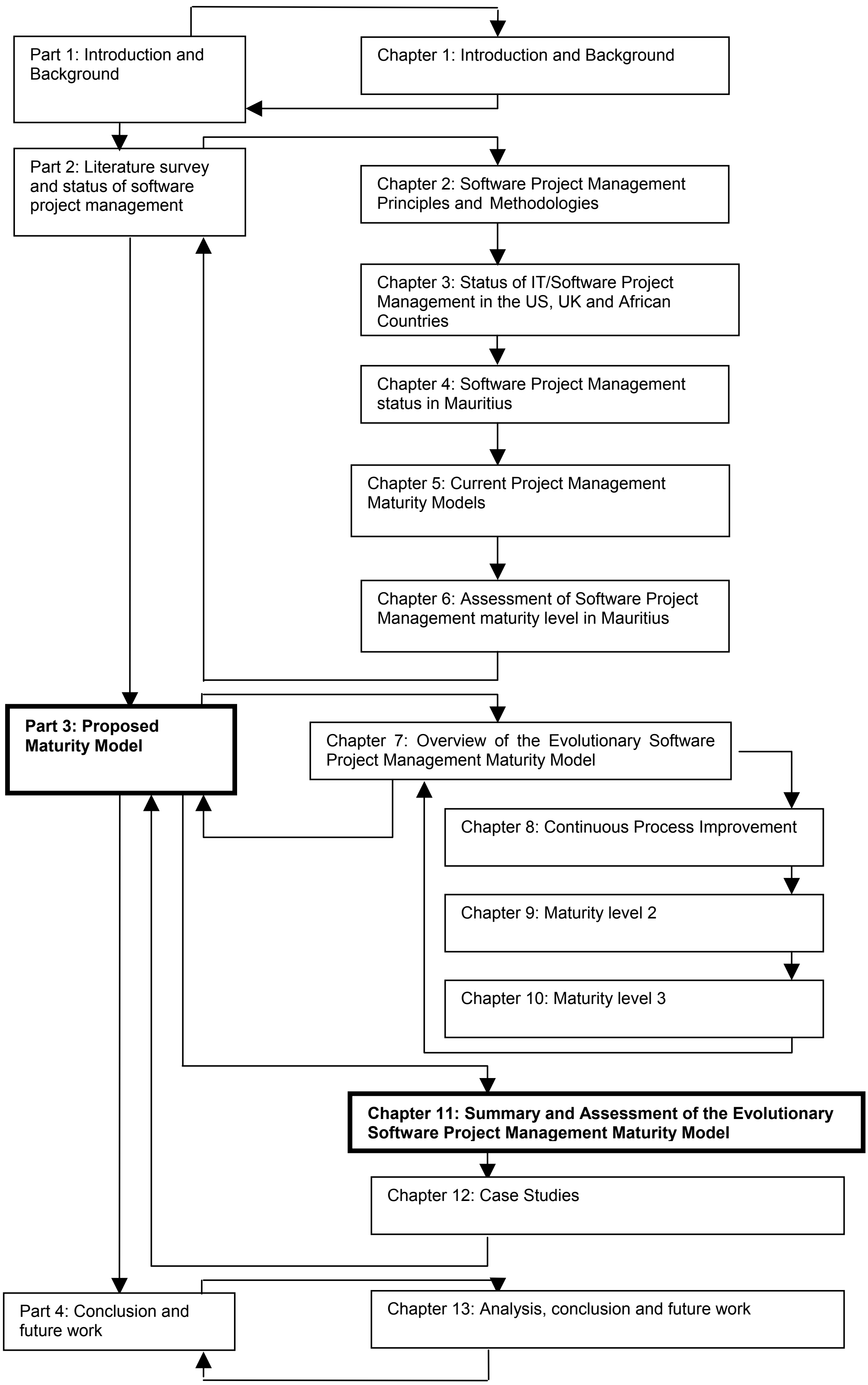


Figure 11.1: Chapter 11 in context within the overall thesis

11.1 Introduction

This chapter summarises the different maturity levels of the ESPM^3 and the continuous process improvement group of KPAs. It helps a reader to see at a glance a description, the features and key process areas within each maturity level. The project manager who has an in-depth knowledge about ESPM^3 may use the summary provided in this chapter as a quick reference guide in order to implement the model for a software project.

In addition, an assessment method is proposed based on a definition of each maturity level. This method is adapted from the research conducted by Sonnekus and Labuschagne (2004).

11.1.1 Aim, objectives and structure of this chapter

In this chapter, a summary and assessment of the ESPM^3 is presented. The assessment provides an organisation with a means to assess its maturity level.

The remainder of the chapter is structured as follows:

Section 11.2 – In this section, the conceptual model is summarised.

Section 11.3 – A summary of the continuous process improvement group of KPAs is presented in this section.

Section 11.4 – Maturity level 1 is briefly discussed in this section. It is noted that this level has no KPAs.

Section 11.5 – In this section, a summary of the maturity level 2 KPAs is presented.

Section 11.6 – The maturity level 3 KPAs are summarised in section 11.6.

Section 11.7 – In this section, assessment of the maturity level of an organisation is explained.

Section 11.8 – In this section, it is noted that the summary in the chapter provides a quick reference guide to project managers. The assessment method for the maturity level of an organisation using ESPM^3 is also highlighted.

11.2 Conceptual representation of ESPM^3

The conceptual model was shown in chapter 7 (also reproduced in figure 10.2) as a conical structure with three maturity levels (levels 1, 2 and 3). A continuous process improvement group of KPAs is also applicable during the software development process. Each maturity level (except for maturity level 1) and the continuous process improvement group comprise a set of KPAs.

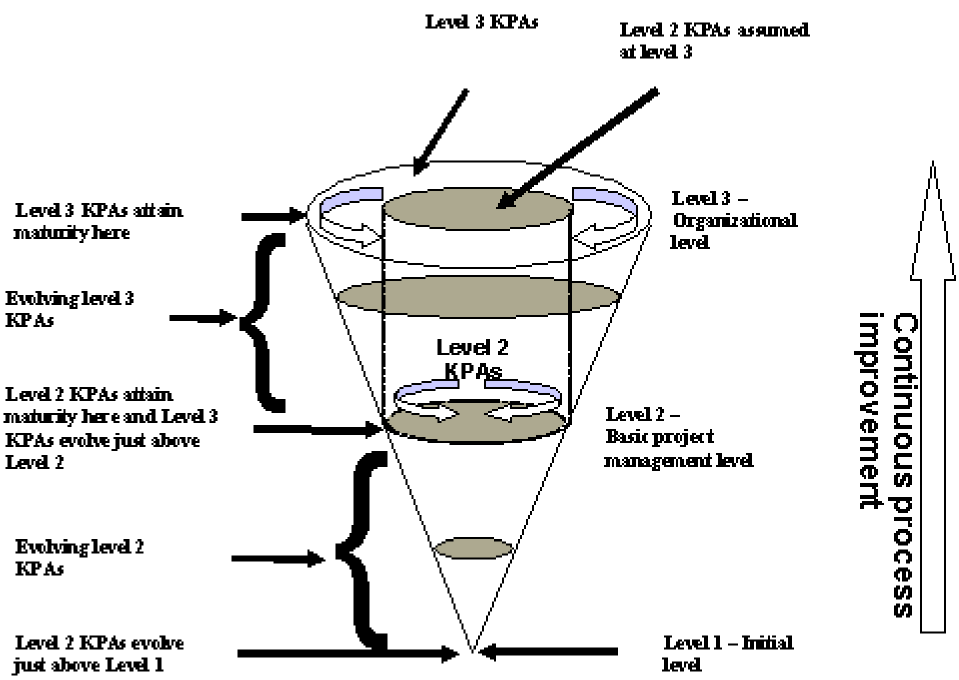


Figure 11.2: Conceptual representation of ESPM^3

Maturity level 1, also known as the initial level, is depicted as the tip of the inverted cone where no KPAs have undergone any development. In other words, maturity level 1 has no KPAs and software development is carried out in an ad-hoc manner. At level 1, the success of any project depends on the skills and experience of the project team.

11.3 Continuous process improvement group

From figure 11.2, the continuous process improvement group of KPAs spans over all the maturity levels except for maturity level 1, where software project management is carried out on an ad-hoc basis. Five KPAs were identified for the continuous process improvement group and they are:

- Soft skills management (KPA_{c1}).
- Change management (KPA_{c2}).
- Software specific focus, (KPA_{c3}).
- Integration management, (KPA_{c4}).
- Environmental management, (KPA_{c5}).

The continuous process improvement group of KPAs, M_c, was depicted as the union of the above KPAs as follows:

$$M_c = \bigcup_{K=1}^{5} KPA_{ck}$$

The characteristics of the continuous process improvement group are summarised as follows:

Continuous process improvement group	
Description	KPAs at this level act at all maturity levels, except the initial level. They influence all other KPAs and adapt to changes in the organisations or the environment.
Key Process Areas	Soft skills management, Change management, Software specific focus, Integration management, Environmental management.
Goals:	
Soft skills management	Satisfied and motivated team members, high performance and productivity.
Change management organisation	Successful change management, improved efficiency.
Software specific focus improved	Software specific management techniques, productivity.
Integration management	Integration of KPAs, successful project results.
Environmental management	Satisfied stakeholders, safe software development environment.

In addition to hard skills, soft skills are used to create a conducive working environment and to motivate team members. Soft skills are considered to be very important for the successful delivery of projects, so much so that Belzer (2004) mentions that projects fail due to the inability of project managers to communicate effectively, work within the organisation's culture, motivate the team, manage stakeholder expectations, understand business objectives and make clear and knowledgeable decisions.

Therefore, ESPM^3 considers the following to be important in the management of software projects:

- Communication skills.
- Team building.
- Creativity.
- Leadership.
- Dissemination of organisational culture.
- Stress management.
- Trustworthiness.
- Conflict management.

Change management takes care of changes in technology, user resistance and other external (like changes in tax figures and legislations) as well as internal changes (like work procedures) impacting on the operation of the organisation. Changes are very common and affect the execution of projects and need to be managed effectively.

A software specific focus is required for the management of software projects. Software projects have certain specificities that make them different from traditional engineering projects. They have strengths that need to be considered and weaknesses that need to be overcomed.

Environmental management is of particular importance in managing software in the Mauritian context. It is known from research that political, social, economic, cultural and legal aspects have to be taken into consideration while managing software projects. Existing methodologies/models/standards in use in developed countries do not incorporate aspects that are important in the developing world. Therefore, ESPM^3 took on board environmental dimensions that are important for software development projects in the Mauritian context.

Integration management ensures that all the KPAs are considered in a holistic and coherent manner so that a common project objective is achieved.

Integration management is, therefore, also important at all maturity levels.

11.4 Maturity level 1

Maturity level 1, also called the initial level, does not possess any KPAs. Software project management is carried out in an ad-hoc or chaotic manner at level 1. Any organisation not using any formal software project management methodology/model/standard is said to be operating at maturity level 1.

Maturity level 1, M_1, was depicted as the empty set, that is:

$$M_1 = \emptyset$$

The characteristics of the maturity level 1 are summarised as follows:

Maturity level 1	Initial level
Description	No KPAs are defined at maturity level 1. Software development processes are carried out in an ad-hoc or chaotic manner and project success depends on the project team. A change in the project team while the project is in progress may disrupt its execution as the processes used and skills are not consistent among project teams.
Key Process Areas	None

11.5 Maturity level 2

Maturity level 2, also called the basic project management level, has three KPAs as follows:

- Time management (KPA_{21})
- Cost management (KPA_{22})
- Quality management (KPA_{23})

Maturity level 2, M_2, was depicted as the union of the above KPAs as follows:

$$M_2 = \bigcup_{j=1}^{3} KPA_{2j} \cup M_c$$

The characteristics of Maturity level 2 are:

Maturity level 2	Basic project management level
Description	Project success is improved by controlling the parameters of time, cost and quality.
Key Process Areas	Time management, Cost management and Quality management
Goals:	
Time management	Work Breakdown Structure, Project Schedule
Cost management	Cost estimates
Quality management	Successful user acceptance, Quality standards (ISO 9001:2000)
Tools and techniques	Gantt charts, Network diagrams

Time management improves the delivery time of a software project. Naturally, a client is interested in knowing when a software product would be in operation. This will help in planning for the change in the organisation to be brought by the new system.

The tracking of cost involved in the development of a software product is a very important activity to be carried out by a project manager to ensure that the cost is under control. The cost is estimated from the effort required to build the software. Any delay in the project can increase the cost to the extent that the profit of the software development organisation may be seriously compromised.

The quality of a software project in $ESPM^3$ refers to the following:

- Functionality.
- Usability.
- Reliability.
- Performance.
- Serviceability.

A software product has to comply with the above criteria to be acceptable to the client. In addition, the software has to be developed within the framework of ISO 9001:2000 in $ESPM^3$ for Mauritius. This ensures that an international quality standard is used for the software development process.

11.6 Maturity level 3

Maturity level 3, also called the organisational level, has three KPAs:

- Human resource management (KPA_{31}).
- Risk management (KPA_{32}).
- Contract management (KPA_{33}).

Maturity level 3, M_3, was depicted as the union of the above KPAs as follows:

$$M_3 = \bigcup_{j=1}^{3} KPA_{3j} \cup M_c$$

The characteristics of Maturity level 3 are:

Maturity level 3	Organisational level
Description	This level ensures consistency of processes across all departments of the organisation.
Key Process Areas	Human resource management, Risk management, Contract management
Goals: Human resource management Risk management Contract management	 An HR plan A Risk management plan A Software contract
Tools and techniques	Responsibility assignment matrix, risk analysis table

As projects are executed by human resources, the effective and efficient management of such resources are key aspects in any software development project. The project manager has to be cautious in selecting team members with the appropriate blend of skills so as to improve the success rate of projects. The team members have to be dealt with using soft skills and at the same time ensuring the availability of the required hard skills.

As all projects involve risks, it is important to anticipate for potential risk hazards and plan for their mitigation and elimination should they occur. All potential risks have to be identified prior to the execution of the project and the ways to deal with them have to be carefully considered.

As most software development tasks involve some contractual obligations, software development organisations have to seek professional advice in order to prepare related contracts. A contract aims to ensure that all parties assume their responsibilities during the execution of the project. Any deviation from the planned outcome has to be dealt with as per the provision of the contract.

11.7 Assessment of maturity level in ESPM^3

ESPM^3 offers a simple method for the assessment of the maturity level of an organisation adapted from the research of Sonnekus and Labuschagne (2004). The assessment method was also reproduced by Sukhoo et al. (2007). Each maturity level is defined in terms of the processes, documentation, management and metrics.

Maturity Level 1 - Initial level

- **Processes** – There is no established practices and standards.
- **Documentation** - Ad-hoc and chaotic.
- **Management** - Management is aware of the need for project management.
- **Metrics** - Collected informally on an ad-hoc basis.

Maturity Level 2 – Basic project management level

- **Processes** - Processes exist, but are not considered to be an organisational standard.
- **Documentation** - Documentation exists on the basic processes.
- **Management** - Management supports the implementation of project management. Projects are executed in a systematic fashion, and management is involved in such projects.
- **Metrics** - Basic metrics to track schedule, cost and quality exist.

Maturity Level 3 - Organisational level

- **Processes** – In addition to basic project management processes, the organisation also considers processes that are applied at an organisational level. These processes involve the clients as members of the project team. All projects use these processes.

- **Documentation** - Documentation exists on all the processes.
- **Management** - Management is regularly involved in the input and approval of key decisions.
- **Metrics** - Metrics are formally collected and each project is evaluated and managed in the light of other projects.

Continuous process improvement is prescribed at all maturity levels, except level 1, by the continuous process improvement group of KPAs. Lessons learnt are also compiled formally at both maturity levels (2 and 3).

Each KPA in ESPM[3] is subject to the Plan-Do-Check-Act process groups, although they are preceded and succeeded respectively by the initiate and the terminate process groups. The maturity level of an organisation is gauged by averaging the observed maturity level of the KPAs of the continuous process improvement group and the expected level reached (either 2 or 3). Table 11.1 is populated based on the definitions above and the average for each column is calculated.

Table 11.1: PDCA cycle for each KPA

Continuous process improvement group	**Plan**	**Do**	**Check**	**Act**
Soft skills management				
Change management				
Software specific focus				
Integration management				
Environmental management				
Maturity level 2				
Time management				
Cost management				
Quality management				
Maturity level 3				
Human resource management				
Risk management				
Contract management				
Average				

Only after an organisation has reached maturity level 2, then it can proceed towards achieving maturity level 3. An assessment exercise has to be conducted in order to ascertain the level reached. The average for each process group can be measured to assess the maturity level of each. An overall maturity level is then computed in order to assess the maturity level of the organisation.

11.8 Conclusion

In this chapter, a concise summary of $ESPM^3$ is presented to allow the project manager who is conversant with the model to have a quick reference for its implementation. Each maturity level was briefly described by highlighting the KPAs and their goals. A summary of the tools and techniques applicable under each maturity level was also given.

In order to allow an organisation to assess its maturity level under $ESPM^3$, a simple assessment method was presented. This method may also provide information on the maturity level of each process group so as to allow the organisation to focus its attention on the specific process group. This assessment method will allow an organisation to pursue its progressive development of an enterprise-wide project management approach, strategy, and decision-making processes.

The next chapter presents two case studies where $ESPM^3$ has been applied. The first case study shows the successful delivery of a project using $ESPM^3$ while the second one reveals a delay in the project where the model is not directly responsible. Therefore, it may be considered that the $ESPM^3$ was successfully applied to both projects.

Chapter 12

Case Studies

Table of contents

12.0 Chapter layout

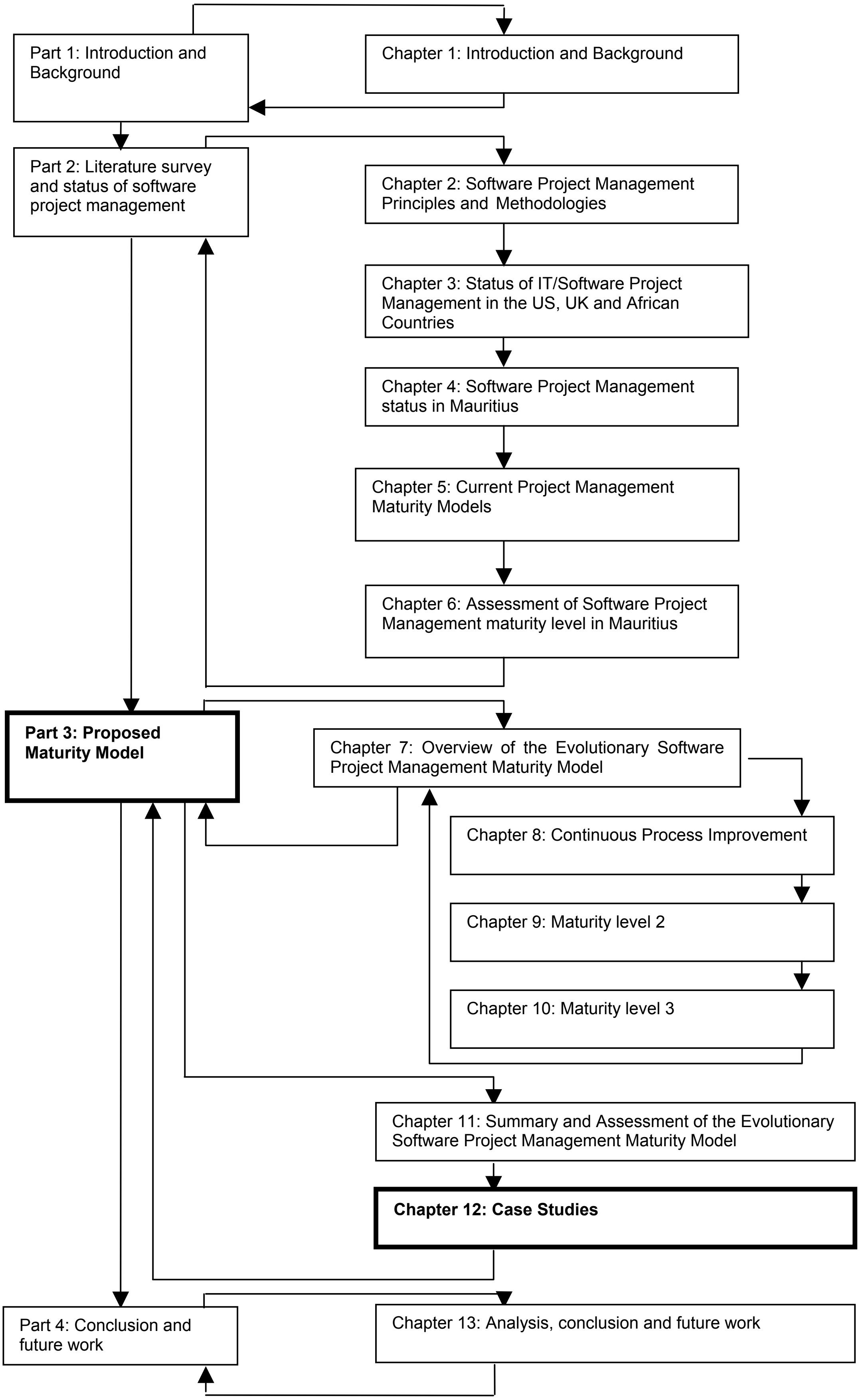


Figure 12.1: Chapter 12 in context within the overall thesis

12.1 Introduction

In this chapter, two case studies are presented. The first case study deals with the development of a *job information system* for the capture and processing of information on jobs as well as the dissemination of relevant information to other organisations and the public. All the KPAs were applied, although an organisation has to reach maturity level 3 to consider all of them.

The second case study involves the implementation of a recruitment system. The project considers the review of an existing software system dealing with the recruitment of staff for various other departments. A web-enabled system covering the computerisation of the recruitment division as well as the administrative activities such as the department's registry and human resources is developed and implemented under $ESPM^3$.

The two software development companies chosen for the case studies have been referred to as ABC and XYZ so as not to reveal any pertinent information about them. In the second case study, the name of the client department concerned has been referred to as "A" for confidentiality reasons. Furthermore, the names of the software systems developed have also been modified for the same purpose.

12.1.1 Aim, objectives and structure of this chapter

In this chapter, two case studies involving the application of the $ESPM^3$ are discussed. All the KPAs were considered to test their applicability although it is recommended for organisations to progress from maturity level 1 to maturity level 3 through maturity level 2.

The remainder of the chapter is structured as follows:

Section 12.2 – In this section, a case study concerning the development of a job information system is discussed. As mentioned above, all the KPAs of the $ESPM^3$ were considered although the organisation concerned was not assessed to be at maturity level 3. It is noted that the project was completed successfully.

Section 12.3 – In this section, another case study was considered for the review of a recruitment system. In this case also, all the KPAs of the ESPM^3 were applied although the organisation concerned was not assessed to be at maturity level 3. This project could not be completed in time as there were problems related to lack of commitment from the users and the problem was not directly related to the model used.

Section 12.4 – In this section, the applicability of the ESPM^3 was ascertained on the basis of the two case studies that were considered.

12.2 Case study 1

Project: Job Information System
Project Description:

Organisation ABC found that it was not able to provide information related to the job market. Such information was scattered at various sites, incomplete, not up-to-date and was not provided through an integrated and appropriate interface. It was also difficult to carry out a job match from information that was received from employers and job seekers.

Organisation ABC, therefore, decided to develop a computerised system to capture and process information on jobs as well as to disseminate relevant information to other organisations and the public. The project also included the development of a website.

ESPM^3 was applied to the software development exercise carried out by the software company. Although the company had its own methodology/model/standard, the project manager agreed to test ESPM^3.

The project comprised the development of a software system and the procurement of the hardware for running the final application. Only the development and implementation of the software was considered for the ESPM[3] model.

The agreed cost, schedule, person's effort were:

Cost: US$ 104,400

Start date/end date: 1 September 2006 to 30 November 2007

Time frame: 31 person months

Implementation of Level 2 KPAs of ESPM[3]

12.2.1 At maturity Level 2

The three KPAs concerned at this level are Time Management, Cost Management and Quality Management.

Time management:

The inputs, processes and outputs of time management are shown in figure 12.2.

Figure 12.2: Time management – Transformation of inputs into outputs

The activities identified were as follows (table 12.1):

Table 12.1: List of activities

Serial No.	Activities
1	Analysis
2	Design
3	Programming
4	Testing
5	Quality control
6	Installation and configuration of software
7	Acceptance testing
8	Training

Experienced staff members were considered for the project as it was of national importance and would bring political, economic and social benefits. It was also approved as a priority project. No prototype was developed, but the experienced users were involved throughout the project.

The project schedule was derived from the list of activities and a Gantt chart, shown in figure 12.3, was prepared.

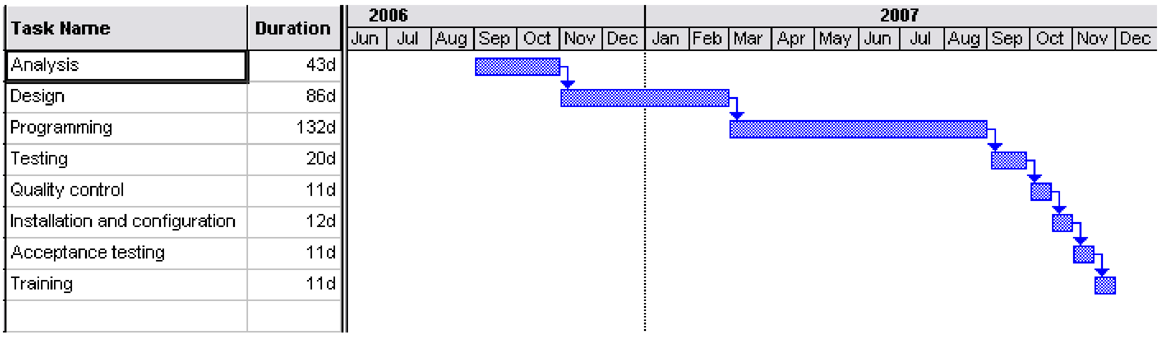


Figure 12.3: Project schedule

Time management was mapped onto four process groups as per the PDCA cycle (plan, do, check and act) shown in figure 12.4.

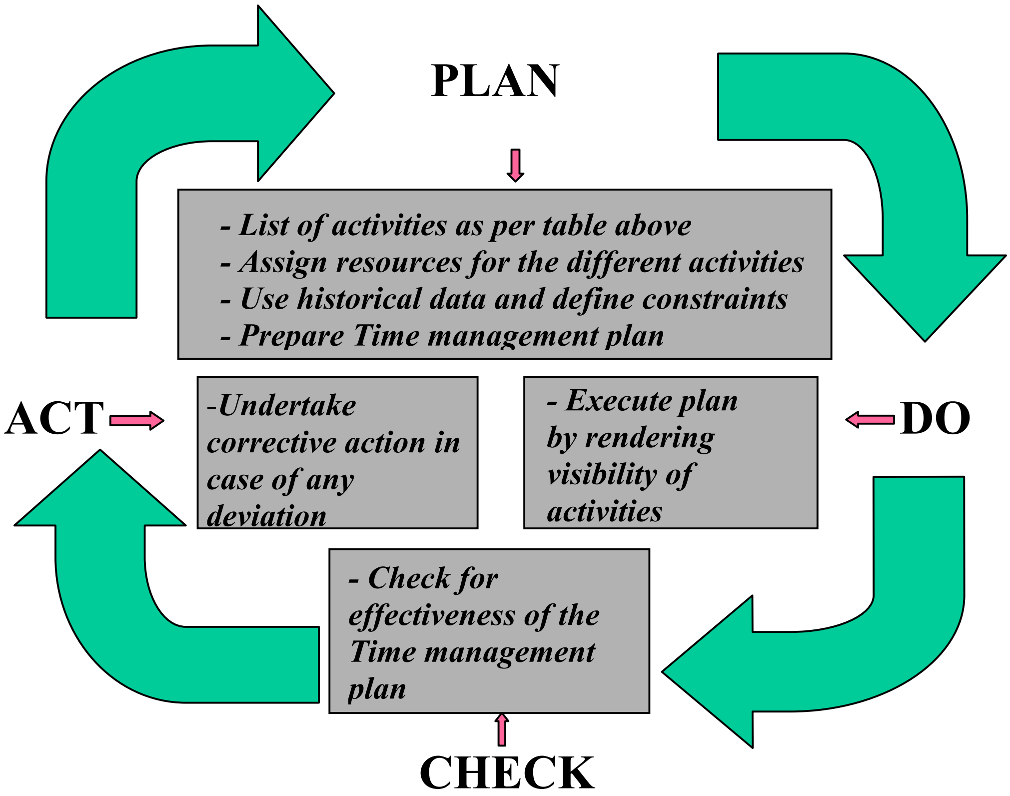


Figure 12.4: PDCA cycle for Time management

Plan: Time management

The time for the different activities was estimated by analogy. The software development company had an extensive experience in analysis, design, testing, quality control and training as per its in-house methodology/model/standard. So, advantage was taken of that experience to estimate the timeframe for these activities.

Do: Time management

As each activity was carried out, a timeline was also drawn to show the planned effort (in blue) and comparison was made with the actual effort (in black). The new Gantt chart showing the planned and actual efforts were as follows:

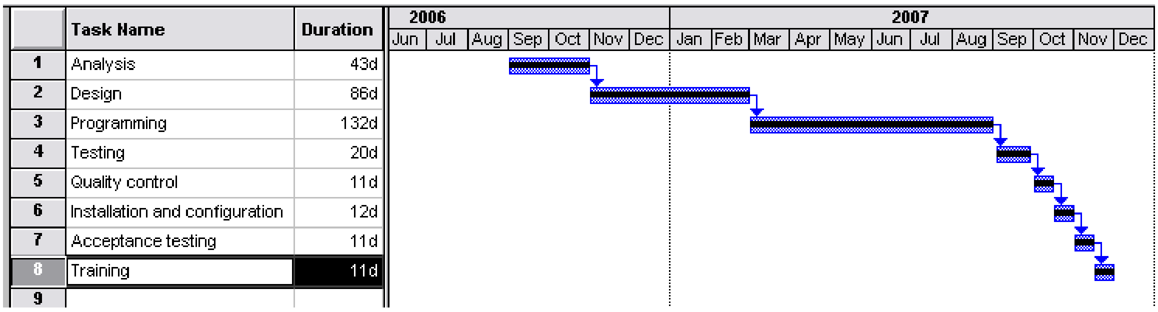


Figure 12.5: Gantt chart for planned and actual efforts

The actual effort was rendered visible using the Gantt chart and the progress of the activities could easily be followed.

Check: Time management

The actual progress of the activities was monitored against the planned activities. The project was executed against a strict time schedule due to its importance. Weekly meetings were held for monitoring purposes. Any tendency for deviation from the planned outcome was immediately rectified and all the activities were executed according to the time schedule.

Act: Time management

No significant action had to be taken for the activities of the project.

Any software error (bug) during the development phase was immediately removed. A certain level of testing was also carried out during the programming activity by the software developer as well as the user. Therefore, errors were corrected at an early phase.

Cost management

A work breakdown structure was developed to reduce the complexity of the software. Resources were assigned for the development of the components of the system. Object points were used to estimate the software development effort required. The transformation of the inputs to the outputs was as shown in figure 12.6.

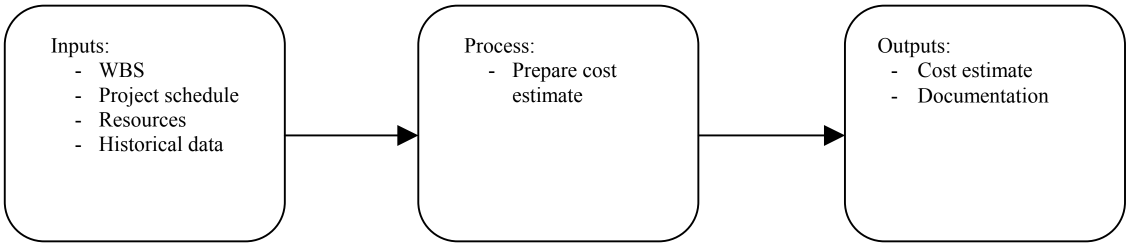


Figure 12.6: Cost management - Transformation of inputs into outputs

Historical data was important in order to estimate the time required to complete the activities identified. The cost estimate was prepared based on the effort required for each activity. The total estimated effort was computed to 31 person-months.

Cost management was mapped onto four process groups as per the PDCA cycle shown in figure 12.7.

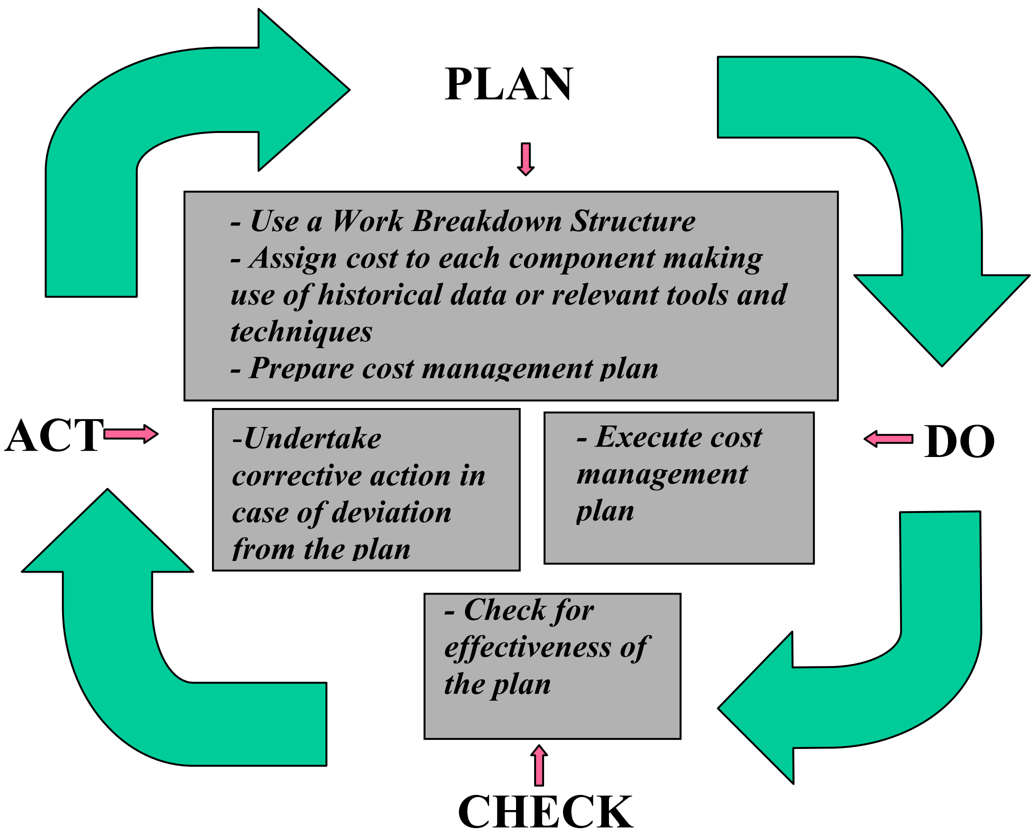


Figure 12.7: PDCA cycle for Cost management

Plan: Cost management

The planning for cost management was carried out from the person-month effort through analogy based on similar previous activities undertaken. In the case of programming, the work breakdown structure helped in subdividing a complex task into simpler tasks to which the object point analysis was applied to estimate the person-month effort.

The total person-month effort was used in order to compute the cost of the project based on the rate applicable at the software development company.

Do: Cost management

The cost was tracked as the project progressed against its schedule. Microsoft Project was used to facilitate the process.

Check: Cost management

The actual cost was compared with the scheduled cost as the project progressed. Microsoft project was used to monitor the cost involved during the progress of the activities of the project.

Act: Cost management

No significant deviation from the planned outcome was noted.

Quality management

The aim of Quality management is to ensure:

- Customer satisfaction.
- Elimination of errors and waste.
- Reduced operating costs.
- Motivation and commitment from employees.

- Profitability and competitiveness.

The inputs of the process generate outputs in the form of:

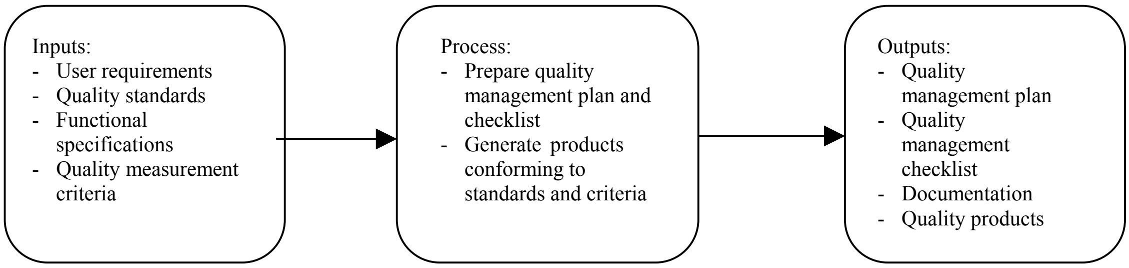


Figure 12.8: Quality management - Transformation of inputs into outputs

Quality management was mapped onto four process groups as per the PDCA cycle shown in figure 12.9.

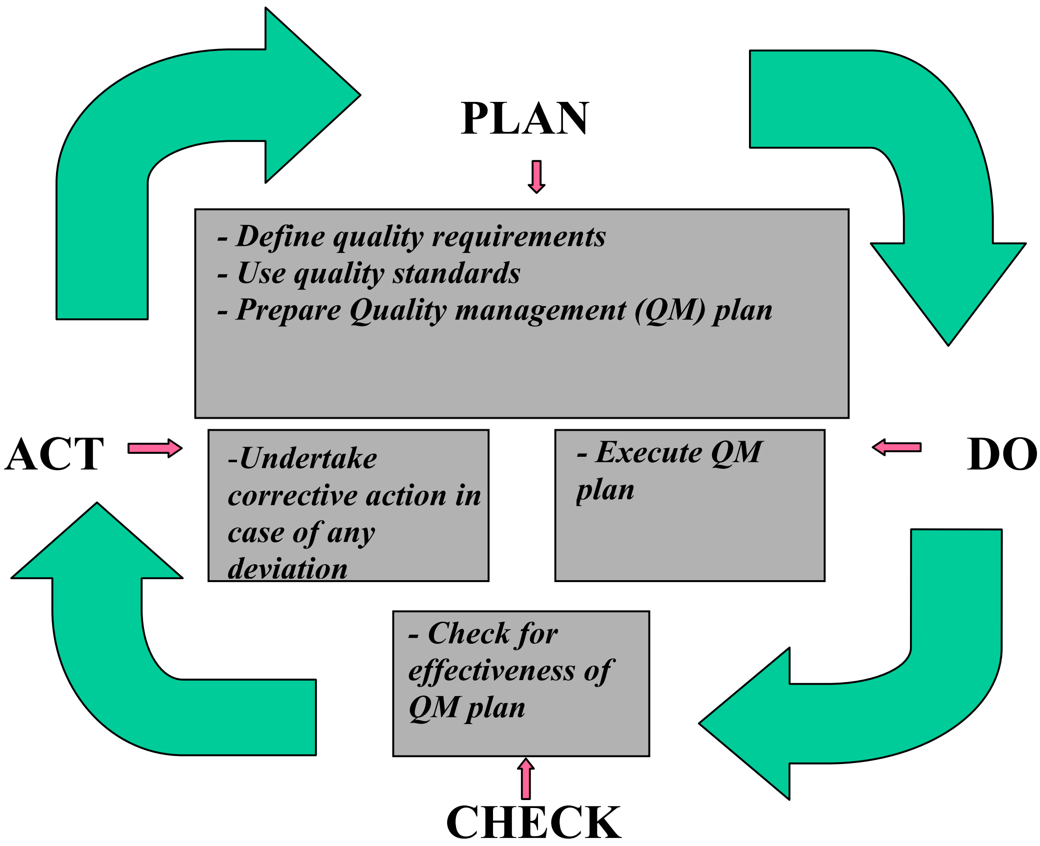


Figure 12.9: PDCA cycle for Quality management

Plan: Quality management

A quality plan was prepared and the quality targets for each deliverable was specified for:

- analysis report.
- systems design report.
- software product.
- documentation.

The analysis and systems design reports were verified by the quality control team for their consistencies, completeness and correctness. An ISO 9001:2000 process was applied in order to ensure consistencies, completeness and to reduce errors. The software product was looked after by a program testing team and quality control staff. In addition, the acceptance test activity was planned to allow the users to test the different functionalities of the software.

Do: Quality management

The plan was executed as per the planned outcome. The software company had an IEEE standard (the IEEE Recommended Practice for Software Specifications (1994 Edition) and the IEEE Recommended Practice for Software Design Descriptions (1994 Edition)) in use. This had the effect of enhancing the quality of the deliverables.

Check: Quality management

The quality of the deliverables was verified against the following criteria:

- Functionality (as defined during the analysis activity).
- Usability.
- Reliability.
- Performance.

- Serviceability.

The acceptance test was considered to be an important activity to ensure that the functionality of the product was met.

Act: Quality management

The quality of the deliverables was regularly checked by the users and amendments were made immediately by the project team. In addition, the developer had to ensure compliance to the IEEE standards. Corrective action was taken as soon as any deviation was detected.

12.2.2 At maturity Level 3

The three KPAs at level 3 are Human Resource Management, Risk Management and Contract Management.

Human Resource Management

Human resource, being the most important resource (Nel et al., 2001) for a project to be executed, was given due consideration. An appropriate blend of skills was taken on board so that the project could be conducted within the agreed time schedule.

In ESPM[3], Human Resource Management (HRM) was carried out using the following inputs:

- Project team attributes, whereby the individual skills were identified and the appropriate blend of skills were taken on board.
- Organisational structure and constraints, whereby the relationships between the staff members were preserved according to the structure prevailing in the organisation and its constraints.
- Employee relations constraints, whereby care was taken so that no employee rights were violated.

In ESPM[3], the outputs were:

- HRM plan detailing the human resources assigned to each activity.
- Documentation on the human resource management plan.

The transformation of the HRM inputs to the outputs is shown in figure 12.10

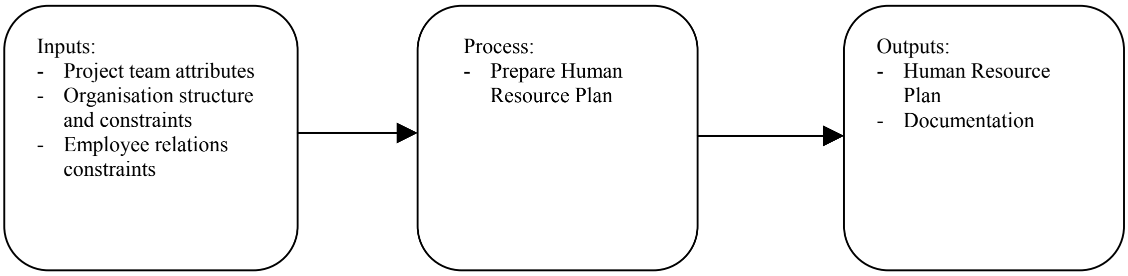


Figure 12.10: HRM - Transformation of inputs into outputs

Human Resource Management was mapped onto four process groups as per the PDCA cycle shown in figure 12.11.

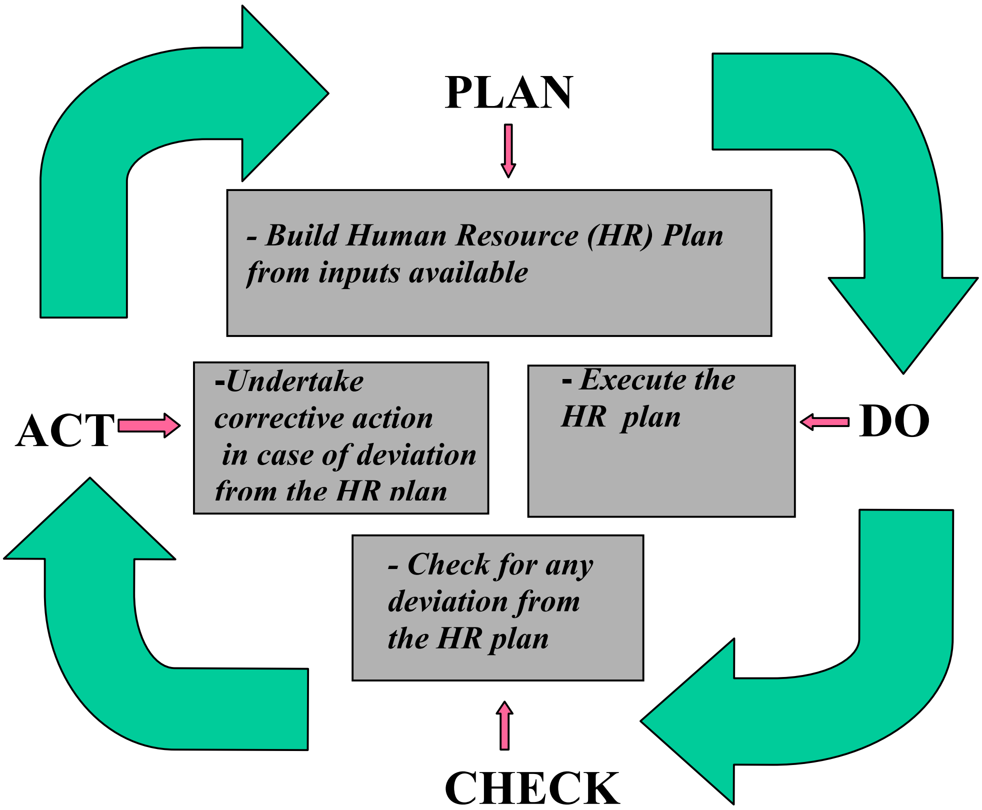


Figure 12.11: PDCA cycle for Human Resource management

Plan: Human Resource Management

The number of resources assigned to the different activities of the project was determined through the planning process group. Effective human resource management is important as it provides for:

- more effective and efficient use of human resources.
- greater employee satisfaction.
- reduced conflicts at work.

Microsoft Project was used to assign resources to the different activities of the project. At a glance, the different resources consumed could be viewed from the Gantt chart as per figure 12.12.

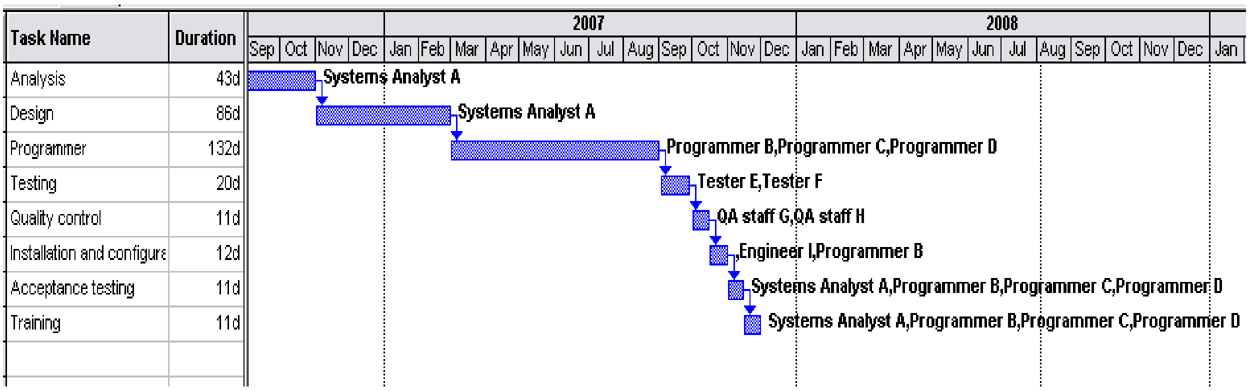


Figure 12.12: Resource allocation

A responsibility assignment matrix was also used to display the resources assigned to each activity. The matrix produced is given in table 12.2.

Table12.2: Responsibility Assignment Matrix

Staff/Phases	A	B	C	D	E	F	G	H	I
Analysis	X								
Design	X								
Coding		X	X	X					
Testing					X	X			
Quality control							X	X	
Installation and configuration		X							X
Acceptance testing	X	X	X	X					
Training	X	X	X	X					

The planning process group also took care of the planning for the selection of project team members and project manager ensured a suitable employee relationship.

Do: Human Resource Management

This process group involved the execution of the human resource management plan by selection of an appropriate team and maintaining an appropriate blend of skills as well as favouring a conducive environment while the project was in progress. The different members of the team were taken on board as per the plan and this was also dependent upon the activities that were executed.

Check: Human Resource Management

The project manager played an important role in monitoring the task performed by each team member and ensured that he/she was working according to the time schedule prepared. Attempts were made to detect and resolve any conflicting behaviour as early on as possible.

No significant conflict, however, occurred during the execution of the project.

Act: Human Resource Management

The different members of the team were brought in at relevant stages to carry out the activities required.

Risk Management

Risk management is concerned with the mitigation or elimination of unforeseen events that may pose a threat to the successful completion of a project. It is achieved by avoiding these risks or drawing up risk management plans for dealing with them. The project manager was responsible for managing the risks that could cause the late delivery of a project, a budget overrun or poor quality software to be developed.

Inputs were identified and the process to transform them into outputs was developed. The following inputs were considered:

- List of potential risks.
- Historical data.

The outputs of the risk management process were:

- Prioritised risks according to their likelihood and impact.
- Risk management plan.
- Documentation.

The transformation of the inputs to the outputs was as per figure 12.13.

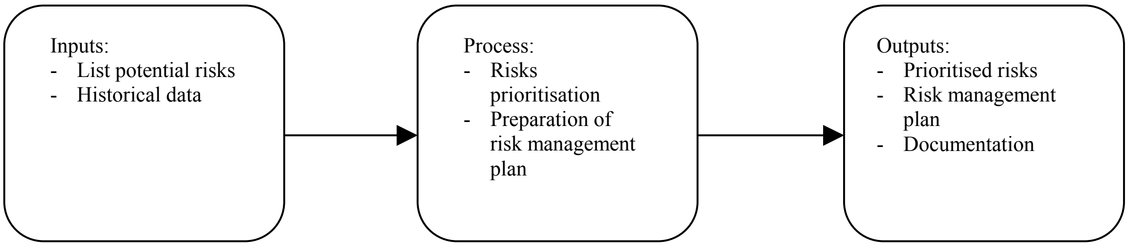


Figure 12.13: Risk management - Transformation of inputs into outputs

Risk management KPA was mapped onto four process groups as per the PDCA cycle shown in figure 12.14.

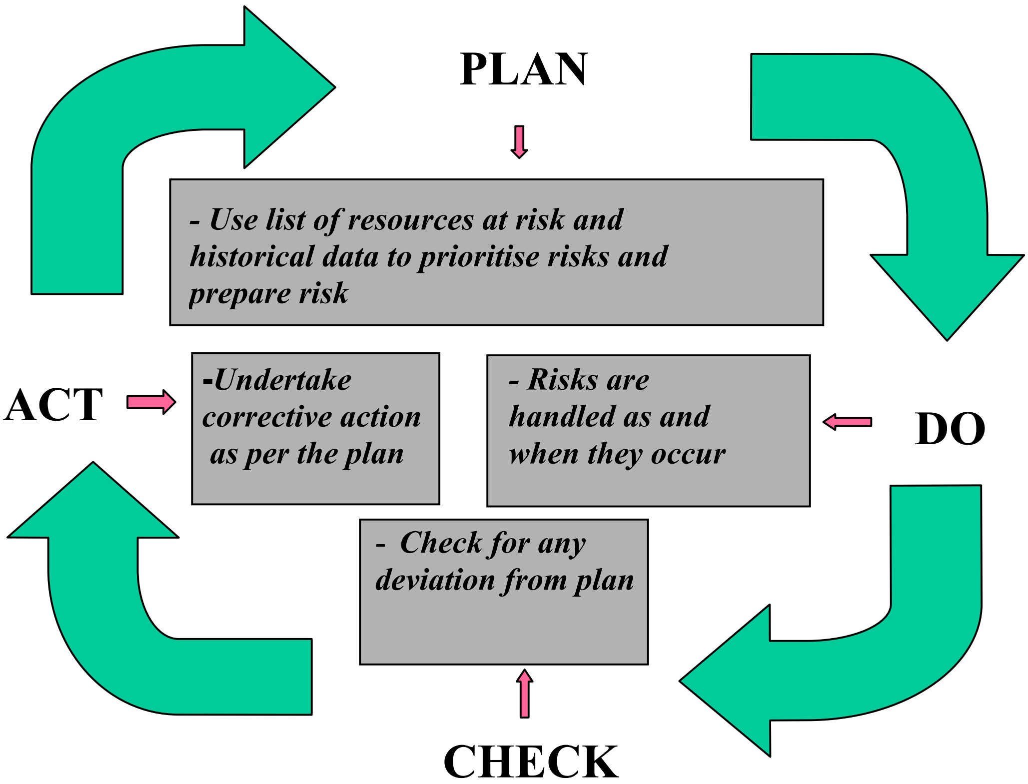


Figure 12.14: PDCA cycle for Risk management

Plan: Risk Management

Those risks that pose a threat to the project were identified, evaluated and prioritised. The potential hazards affecting the project resources were listed. Values between 1 and 5 were assigned to the risks identified as follows:

1 = No impact

2 = Low impact

3 = Moderate impact

4 = High impact

5 = Very high impact

Similarly, the likelihood for the occurrence of an identified risk hazard was also assigned a value from 1 to 5.

The risk analysis table prepared is shown in table 12.3.

Table 12.3: Risk analysis table

Risk	**Likelihood**	**Impact**	**Risk exposure**
Changes to specifications	1	4	4
A key staff being ill or leaving the job during critical activities	1	3	3
Analysis taking more time than planned	2	4	8
Design taking more time than planned	2	4	8
Coding taking more time than planned	2	4	8
Users not satisfied with the quality of the software in terms of functionality	1	5	5

Risk exposure is the product of the likelihood of a risk hazard occurring and its impact on the successful delivery of the project (see section 10.4.1).

The risks were then prioritised in terms of their risk exposure values. A new table (table 12.4), sorted in descending values of risk exposure, was generated.

Table 12.4: Risk exposure table

Risk	Likelihood	Impact	Risk exposure
Analysis taking more time than planned.	2	4	8
Design taking more time than planned.	2	4	8
Coding taking more time than planned.	2	4	8
Users not satisfied with the quality of the software in terms of functionality.	1	5	5
Changes to specifications.	1	4	4
A key staff being ill or leaving the job during critical activities.	1	3	3
Client site not ready for acceptance test and training.	4	4	16

The project manager then gave special attention to the high risk exposure values.

A risk management plan (table 12.5), elaborating on the means to deal with each risk hazard, was also prepared as below and the plan allowed the project manager to adopt a proactive measure to deal with risks instead of adopting a crisis management attitude.

Table 12.5: Risk management table

Risk	Likelihood	Impact	Risk exposure	Method to deal with the risk hazard
Client site not ready for acceptance test and training.	4	4	16	The acceptance test and training can be conducted at the premises of the software development company.
Analysis taking more time than planned.	2	4	8	Overtime hours to be considered.
Design taking more time than planned.	2	4	8	Overtime hours to be considered.
Coding taking more time than planned.	2	4	8	Adding more people on the activity. Given that the software was to be developed in modules and each module had its own specifications that were prepared by the systems analysts, it would be plausible for any programmer of the software development organisation to be assigned certain tasks.
Users not satisfied with the quality of the software in terms of functionality.	1	5	5	Testing and quality control to be effected by experienced staff.
Changes to specifications.	1	4	4	The user has to bear the additional cost according to the contract. Some changes may also be considered as a review exercise if possible.
A key staff member being ill or leaving the job during critical activities.	1	3	3	Replacement staff was identified.

Do: Risk management

Risk management was carried out using the risk management plan developed in the “Plan” process group (table 12.5).

Check: Risk management

This process group allowed the project manager to monitor the occurrence of any risk hazards. With the help of the risk management plan, the occurrence of each potential risk hazard was identified as soon as it occurred for subsequent action. No major deviation from the planned outcomes was noted.

Act: Risk management

There was no significant action taken in order to counteract risk hazards for this project.

Contract management

A contract, being a formal agreement between two parties, obligates one party (the software development organisation) to perform a service and the other party (the client) obligates itself to do something in return, usually in the form of a payment (see section 10.5). In the case of the project, it was a direct procurement exercise as it was a review of an existing system whereby the supplier of the existing system was requested to carry out the review exercise.

For the project, contract management made use of the following inputs:

- Requirements prepared by the client.
- Proposal from the contractor.
- Clauses in the contract and deliverables.

The output generated was:

- Successful contract execution.
- Acceptance of the deliverables by the client

The transformation of the inputs to the outputs for contract management is shown in figure 12.5.

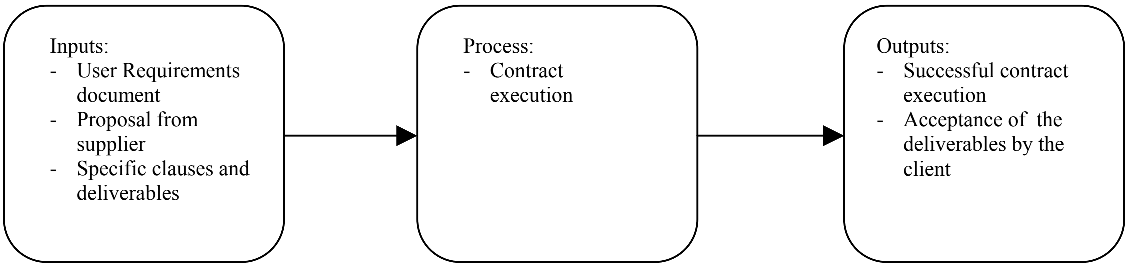


Figure 12.15: Contract management - Transformation of inputs into outputs

The project manager was aware of the implications of the clauses in the contract and he ensured as far as possible for compliance unless there was mutual agreement of performing the task otherwise.

The mapping of contract management onto the four process groups as per the PDCA cycle is shown in figure 12.16.

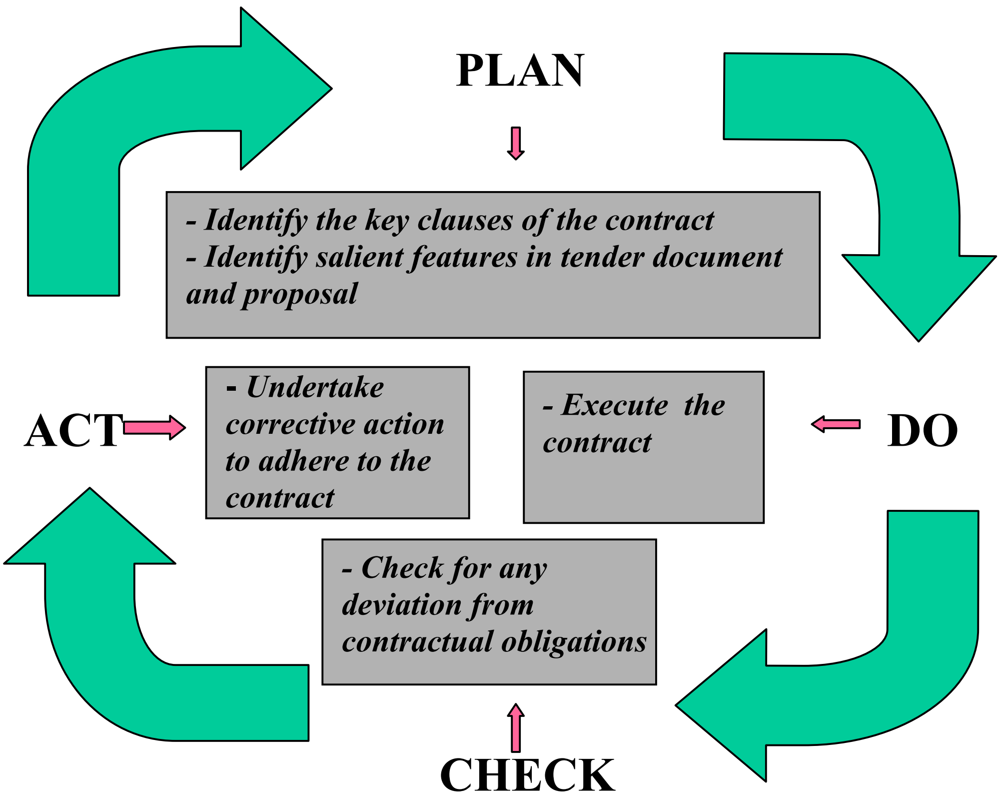


Figure 12.16: PDCA cycle for Contract management

Plan: Contract management

The services of a legal advisor were sought for the preparation of the contract. Upon signature of the contract, the project was started. In the contract, the parties concerned, the work to be carried out, the terms of payment, the deliverables to be submitted at a specified period of time were specified, in addition to other clauses like the extension of the contract in the case of act of God, cancellation of the contract and penalties to be applied in the case of non respect of the contractual agreements.

Work around solutions were anticipated in case of problems occurring during the execution of the project. For this project, it was planned for acceptance testing and training to be carried out at the premises of the software development organisation if the client's site was not ready as per table 12.5.

Do: Contract management

Upon signing of the contract, a percentage of the contract value was claimed from the client in order to start the project. All deliverables that had to be provided were noted together with their dates against the plan.

A list of important issues in the contract was prepared by the project manager to deal with all the clauses forming part of the contract during the execution of the project.

Check: Contract management

The deliverables were checked regularly against the plan to ensure their timely delivery. No deviation from the planned outcome was noted. The schedule of activities was verified against the plan on a weekly basis by the members of the steering committee.

Act: Contract management

As all the activities were kept under close control, no action was necessary for the contract compliance.

12.2.3 Continuous Process Improvement group of KPAs

The five KPAs concerned here are Soft Skills Management, Change Management, Software Specific focus, Environmental Management and Integration Management.

Soft Skills Management

Soft skills, concerned with the management and working with people, were applied to this project. Soft skills management was carried out through the following inputs that were transformed into outputs:

- List of soft skills.
- Soft skills remedial action knowledge base.
- Historical data.

The transformation yielded the following outputs:

- Satisfied team members.
- High performance and productivity.
- Lessons learnt to be utilised for future soft skills management.

The transformation of the inputs to the outputs is given in figure 12.17.

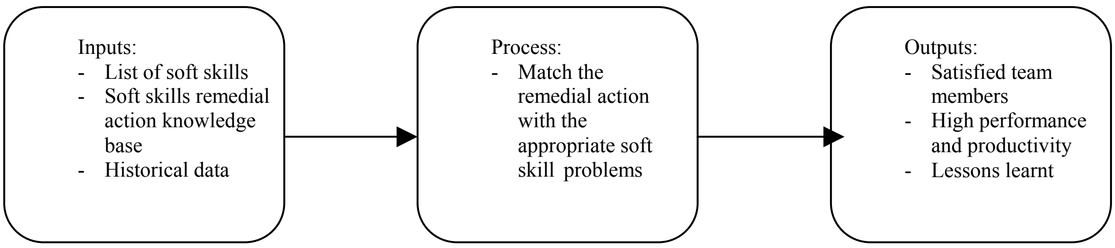


Figure 12.17: Soft skills management - Transformation of inputs into outputs

The PDCA cycle for Soft Skills Management

Soft skills management was mapped onto four process groups as per the PDCA cycle shown in figure 12.18.

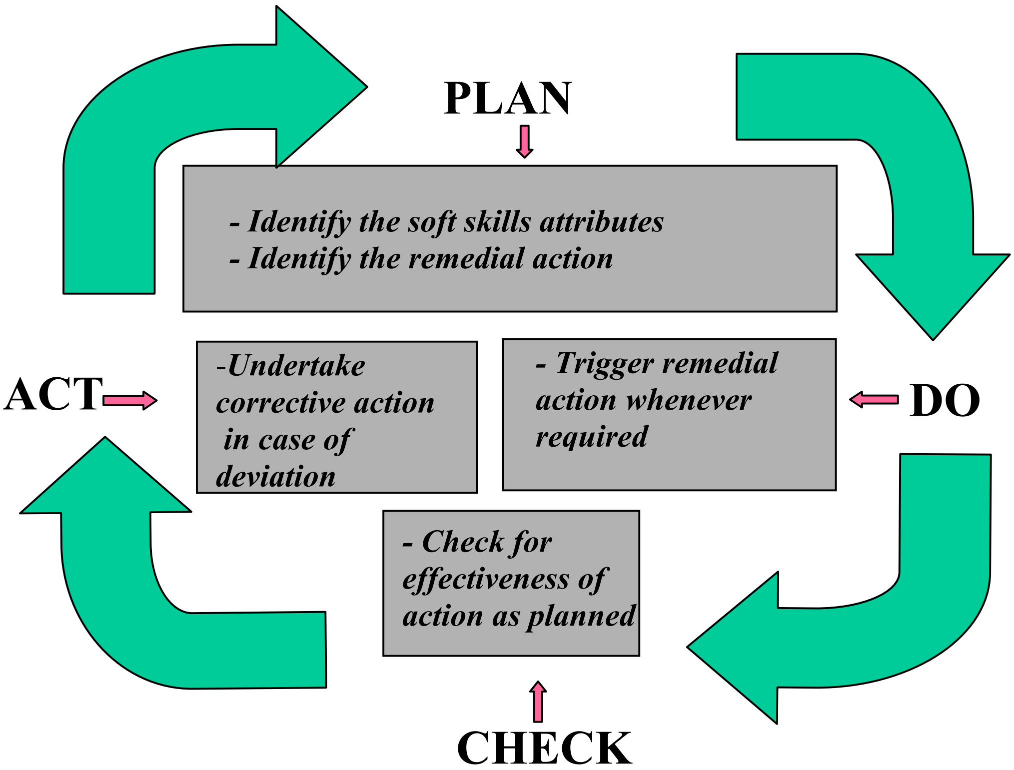


Figure 12.18: PDCA cycle for soft skills management

The soft skills considered were Communication skills, Team building, Creativity, Leadership, Organisational culture dissemination, Stress management, Trustworthiness and Conflict management.

Plan: Soft Skills management

The planning process group is intended to be the stage for listing all the possible remedial actions to be taken to deal with certain shortcomings arising from the management of staff. This is shown in table 12.6.

Table 12.6: Soft skills management table

Problem requiring soft skills management	**Soft skills attribute**	**Possible remedial actions**	**Appropriate action for the project**
Inadequate Communication	Communication skills	Honest discussion.	
		Listen to person concerned.	
		Negotiate.	
		Facilitate discussion.	
Inappropriate mix of skills	Team building	Swapping team members between teams.	
		Participation of a member from another team to provide services on a part time basis.	
Lack of user-friendliness of software	Creativity	Feedback from users.	
		Use of standards developed within the organisation.	
De-motivated staff	Leadership	Consensus building through listening and understanding the reasons for de-motivation.	
		Use of appropriate leadership style.	
		Provision of appropriate incentives and rewards.	
Opposing organisational culture	Organisational culture dissemination	Holding discussion to facilitate commitment.	
		Provide information on the need for the prevailing culture for efficiency in the organisation.	
Excessive stress originating from the need to ensure delivery of project within time, cost and quality constraints	Stress management	Regular feedback on the status of all aspects of a project and providing any corrective action.	
		Use of appropriate communication channels for the reporting of problems pertaining to delivery of project so that corrective action is taken.	
Lack of trust	Trustworthiness	Discussion, listening and understanding.	
		Consensus building to enhance trust.	
Conflict between team members	Conflict management	Mediation.	
		Arbitration.	
		Control.	
		Acceptance.	
		Elimination.	

Do: Soft Skills management

During the execution of the project, the relevant soft skills were identified and the appropriate actions were taken as per table 12.7 prepared during the planning phase. In the table 12.7, the relevant action was marked with a tick (√).

Table 12.7: Execution of soft skills

Problem requiring soft skills management	Soft skills attribute	Possible remedial actions	Appropriate action for the project
Inadequate Communication	Communication skills	Honest discussion	
		Listen to person concerned	
		Negotiate	
		Facilitate discussion	√ The Project Manager facilitated discussion between the user and the project team during the execution of the project in order to obtain full commitment of the users.
Inappropriate mix of skills	Team building	Swapping team members between teams	
		Participation of a member from another team to provide services on a part time basis	
Lack of user-friendliness of software	Creativity	Feedback from users	
		Use of standards developed within the organisation	
De-motivated staff	Leadership	Consensus building through listening and understanding the reasons for de-motivation	
		Use of appropriate leadership style	
		Provision of appropriate incentives and rewards	
Opposing organisational culture	Organisational culture dissemination	Conducting discussions to facilitate commitment	
		Provide information on the need for the prevailing culture for efficiency in the organisation	√ By including the user in the team, it was necessary to resolve conflicting culture between the contractor's and user's organisations. For example, working beyond normal office hours was a common practice for the contractor but users could not be convinced to extend their working hours accordingly.
Excessive stress originating from aneed to ensure delivery of project within time, cost and quality constraints	Stress management	Regular feedback on status of all aspects of a project and providing any corrective action	√ The stress to complete the project on time started to pick up as from the late stages of the programming activity. The Project Manager played an active role in motivating the team and relieving the stress.
		Use of appropriate communication channel for reporting of problems pertaining to delivery of project so that corrective action is taken	
Lack of trust	Trustworthiness	Discussion, listening and understanding	
		Consensus building to enhance trust	
Conflict between team members	Conflict management	Mediation	
		Arbitration	
		Control	
		Acceptance	
		Elimination	

Given the fact that a small team undertook the development of the project, the minimum soft skills attributes were found to be necessary. However, the good communication skills among the members of the project team further promoted conducive working relationships.

Check: Soft Skills management

The project manager instilled good communications among the project team. Right at the start of the project, the project manager also ensured that an appropriate blend of skills was available.

Act: Soft Skills management

The use of soft skills ensured good communications between all the members and no conflicts resulted during the execution of the project.

The details on change management, software specific focus, environmental management and integration management regarding the project are found at appendix D.

Lessons that could be learnt from the project were:

(i) Experienced staff members were considered for the project. They contributed towards the successful delivery of the project.

(ii) User involvement throughout the project contributed to user commitment and the elimination of bugs during the early stages of software development.

(iii) The political, social and economic benefits to be reaped also resulted in commitments at all levels (Software development company, user, management) to ensure that the software product was developed within the time specified, on budget and according to the quality level specified.

The project was completed within the specified period given that ESPM^3 focused on the political and social nature of the project (through the Environmental KPA), soft skills, risk management as well as strengths and weaknesses associated with software development. In addition, other aspects (or KPAs) commonly employed by other methodologies/models/standards were considered. Given the importance of the project, staff members with extensive experience were assigned to the project. It was necessary to support and encourage the development team throughout the execution of the project. Risks were kept under control through the relevant KPA in the model. It was noted that the use of ESPM^3 contributed towards the success of the project through its KPAs. The model showed its suitability for the project of a software nature as well as an environment with constraints specific to Mauritius, being an instance of the set of developing countries.

12.3 Case study 2

Project: Recruitment systems review

Project Description:

The existing host-based system at department A was in use since 1990. In order to adapt to changes in the department and at the same time reap the benefits of new technology, it was decided to review the system and include new functionalities like e-services so as to enable members of the public to submit application for jobs online.

Therefore, the project for the review of the system comprised a review of the existing software dealing with recruitment of staff for various other departments including department A. The new web-enabled system covered the computerisation of the recruitment division as well as the administrative activities such as the department's registry and human resources. The review of the application software also necessitated a replacement of the existing hardware that was considered to be an obsolete technology.

The same company (XYZ) that developed and provided maintenance services for the original software was entrusted the review exercise. This ensured a minimum delay in gathering requirements as the company was already aware about the activities in the department. The latest development (at the time of awarding the contract) platform was chosen by the software company.

The project comprised the tasks encompassing analysis, design, software development, implementation and training. The company responsible for the development of the software and its implementation was already operating at level 3 of the CMMI.

Cost: US$ 118,500

Proposed start date/end date: April 2006 to November 2007

Time frame: 30 person-months

Implementation of ESPM^3

12.3.1 At maturity Level 2

The three KPAs concerned here are Time Management, Cost Management and Quality Management.

Time management:

The transformation of the inputs to the outputs is shown in figure 12.19.

Figure 12.19: Time management - Transformation of inputs into outputs

The activities identified were analysis, design, programming, testing and quality control, implementation and quality control. The detailed list prepared was:

Table 12.8: List of activities

Serial No.	Activities
1	Analysis
2	Design
3	Development and demonstration of prototype
4	Programming
5	Testing
6	Quality control
7	Installation and configuration of software
8	Acceptance testing
9	Migration of data
10	Training
11	Live run

Given that there was a tight deadline, the absence of any key staff member was foreseen as a source of delay. It was, therefore, necessary to consider this as a risk that was taken care of by the risk management key process area.

Resources included human resources, namely the analyst, programmer, the quality control team and administrative staff.

The historical data considered was that the key staff had enough experience as they were the staff involved in the initial development of the software. There were not much learning processes as regard to reading and understanding the technical documentation on the software although they were prepared a long time back. The company also had a project management model in place (CMM followed by CMMI) and was applied by all staff. In addition, all problems faced during previous projects were well documented and all staff members were made aware of such documentations. These lessons learnt were helpful to deal proactively with problems or to keep risks under control.

The project schedule was derived from the list of activities and a Gantt chart shown in figure 12.20 was prepared.

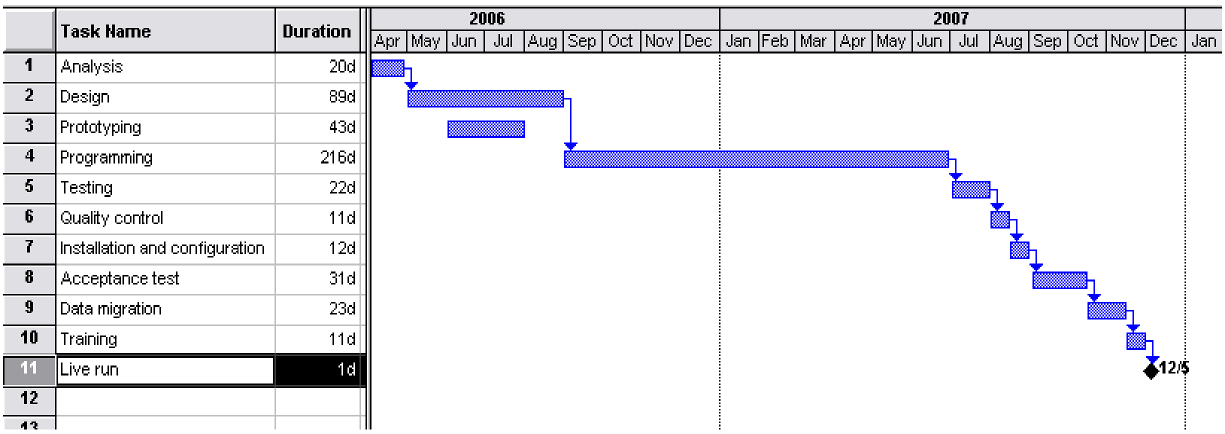


Figure 12.20: Gantt chart showing list of activities

The mapping of time management onto the four process groups as per the PDCA cycle shown in figure 12.21.

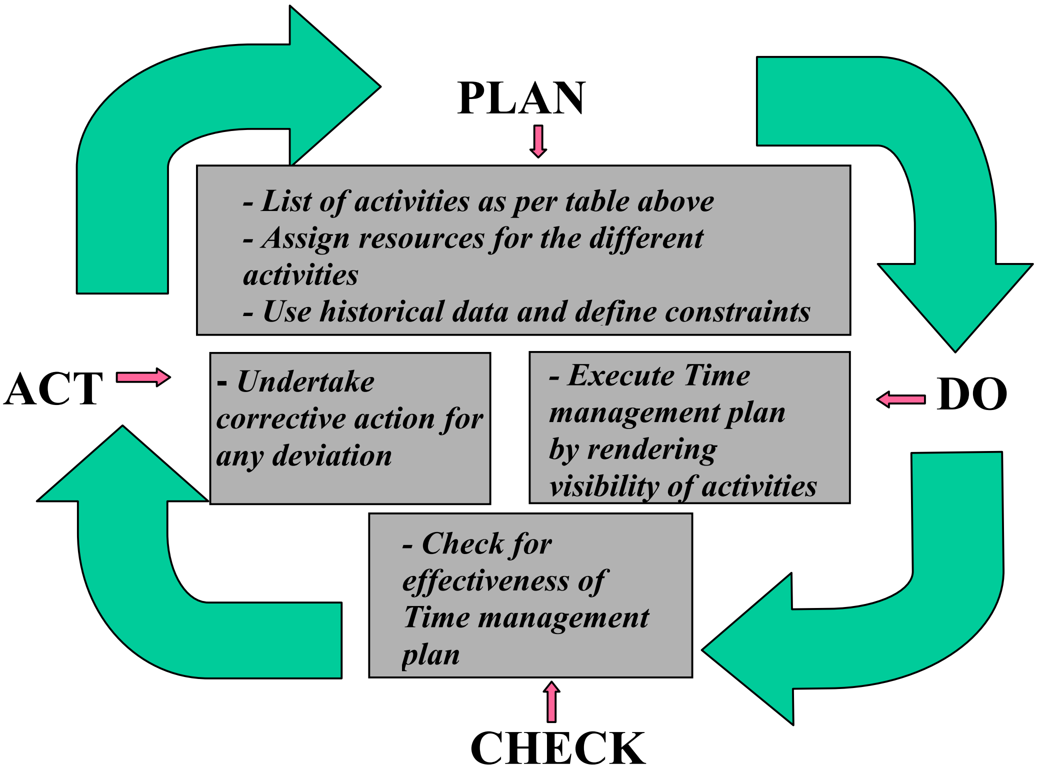


Figure 12.21: PDCA cycle for Time management

Plan: Time management

The times for the different activities (except for programming) were estimated by analogy. The software development company has an extensive experience in analysis, design, testing, quality control and training as per its CMMI model in place. So, advantage was taken of this experience to estimate the timeframe for these activities.

Do: Time management

As each activity was carried out, a timeline was drawn alongside the planned effort and comparison was made with the actual effort. The new Gantt chart showing the planned and actual efforts (with the first row for each activity showing the planned effort while the second row showing the actual effort) appears in figure 12.22.

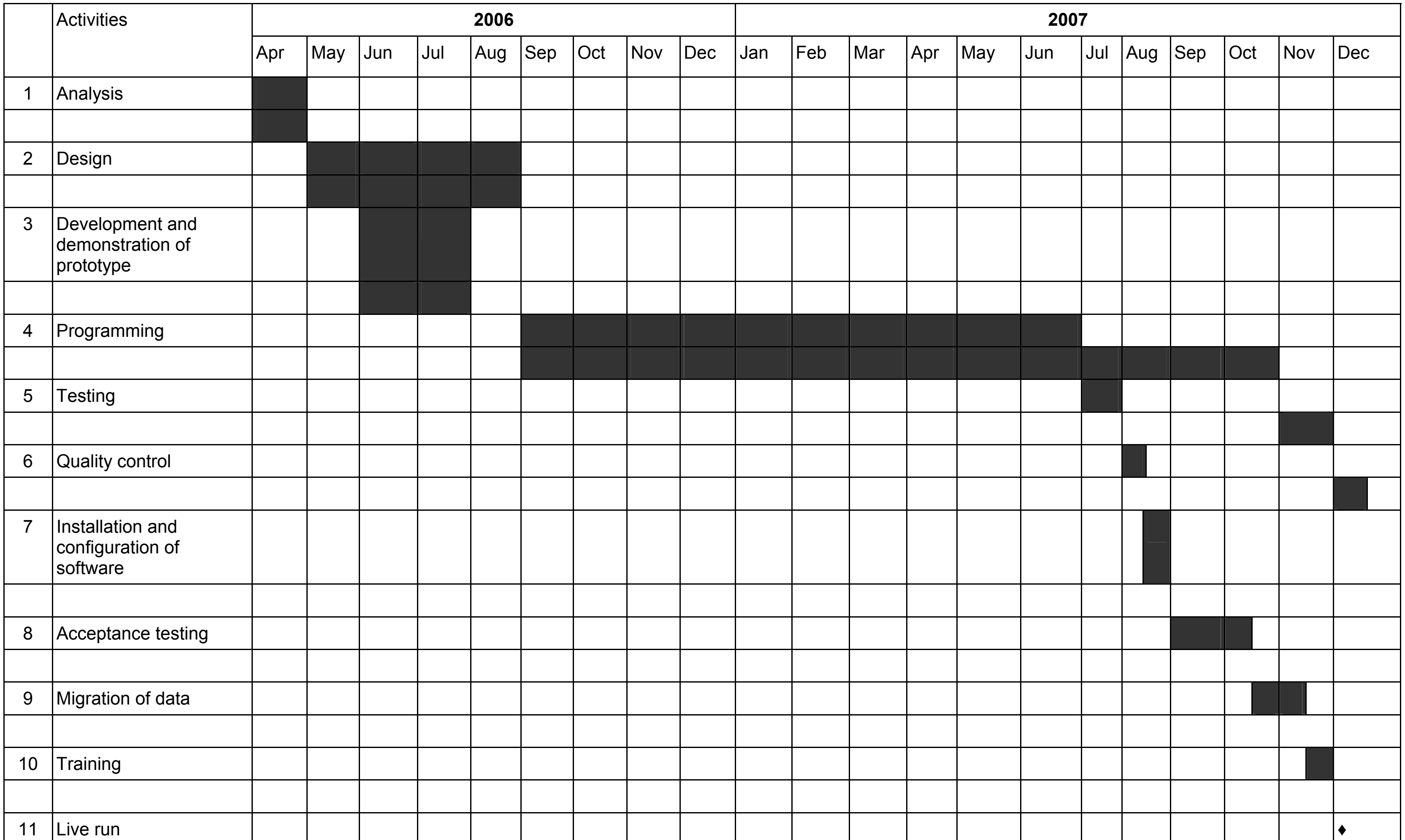


Figure 12.22: Gantt chart showing each activity of the project

The actual efforts were rendered visible using the Gantt chart and the progress of the activities could easily be followed.

Check: Time management

The actual progress of the activities was monitored against the planned activities. A deviation from the planned efforts was noted during the programming activity. In fact, the programming activity was completed four months later. This was due to the use of a latest version of the database management system (DBMS) while the development of the software was already underway with a previous version. The staff had no prior experience with the new version of the database management system and the user was also not agreeable to continue with the development of a software product using an old version of the DBMS as support for the old version would no longer be available.

Consequently, all the other activities were delayed. Further delay was encountered with the installation and configuration activity as there were major disturbances owing to construction works and the equipment were to be relocated in a new computer room. The users felt that it was not the appropriate time to perform the acceptance test.

ESPM[3] could have helped in this problem by facilitating the exposure of the staff members to the forthcoming technologies through training. Normally, beta versions are available prior to the release of a new DBMS. The staff involved in software development and database administration could be provided with the required training through the use of beta versions of a DBMS.

Act: Time management

Training was nonetheless conducted at the software developer's site so as to minimise the delay as far as possible and also abide by the contractual agreement (see section 10.5.2). The acceptance test was only possible as from mid January 2008. The Gantt chart had to be revised. Due to a lack of user commitment, the acceptance testing could not be completed successfully by end of February 2008. With a lot of difficulties, the software development organisation managed to obtain an agreement in principle that the software product was accepted by department A. The software development organisation submitted an invoice for payment to be effected in June 2008 and this triggered the management of department A to request the IT staff concerned to certify in writing that the software complied to their requirements.

The IT staff requested the different sections to check the functionality of the software again and several errors were identified. The software developer was called to attend to the problems, but other errors were noted. The only programmer on the project left the organisation at that critical time and had to be replaced by another one. This was an unforeseen event and not much could be done to remedy the solution immediately (Hughes and Cotterell, 2006). Not much time was spent in acquiring the necessary information on the project as a proper handling over was done and the documentation on the systems was properly kept. Both parties agreed to a joint responsibility for the problem. A new extended time schedule (till the end of September 2008) was prepared to carry out the acceptance testing and to eliminate the errors found.

Cost management

Each activity was assigned the required resources. A work breakdown structure was prepared for the different components of the system and resources were assigned to them. Object points were used to estimate the software development effort required. The transformation of the inputs to the outputs is shown in figure 12.23.

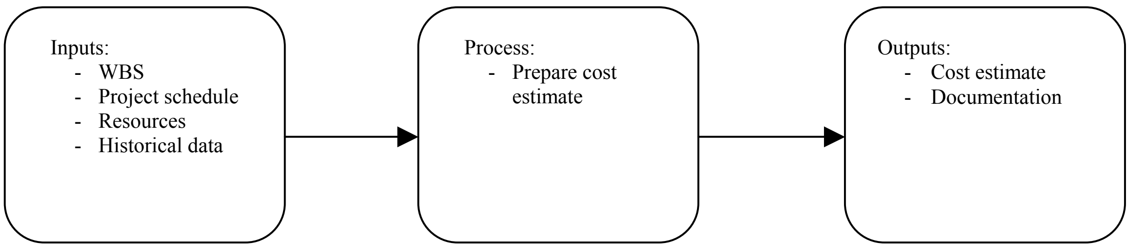


Figure 12.23: Cost management - Transformation of inputs into outputs

The time and cost management KPAs are interlinked as the person-month effort was used to prepare the project schedule. As a result, Function point analysis, COCOMO and Object point analysis may be used to calculate the cost (Schach, 2002; Hughes and Cotterell, 2006). The historical data was important in order to estimate the time required to complete the activities identified. The cost estimate was prepared based on the effort required for each activity. The total estimated effort was computed as 30 person-months. Programmers were assigned to provide 20 person-months effort during the development of the prototype and the programming phase.

Cost management was mapped onto four process groups as per the PDCA cycle shown in figure 12.24.

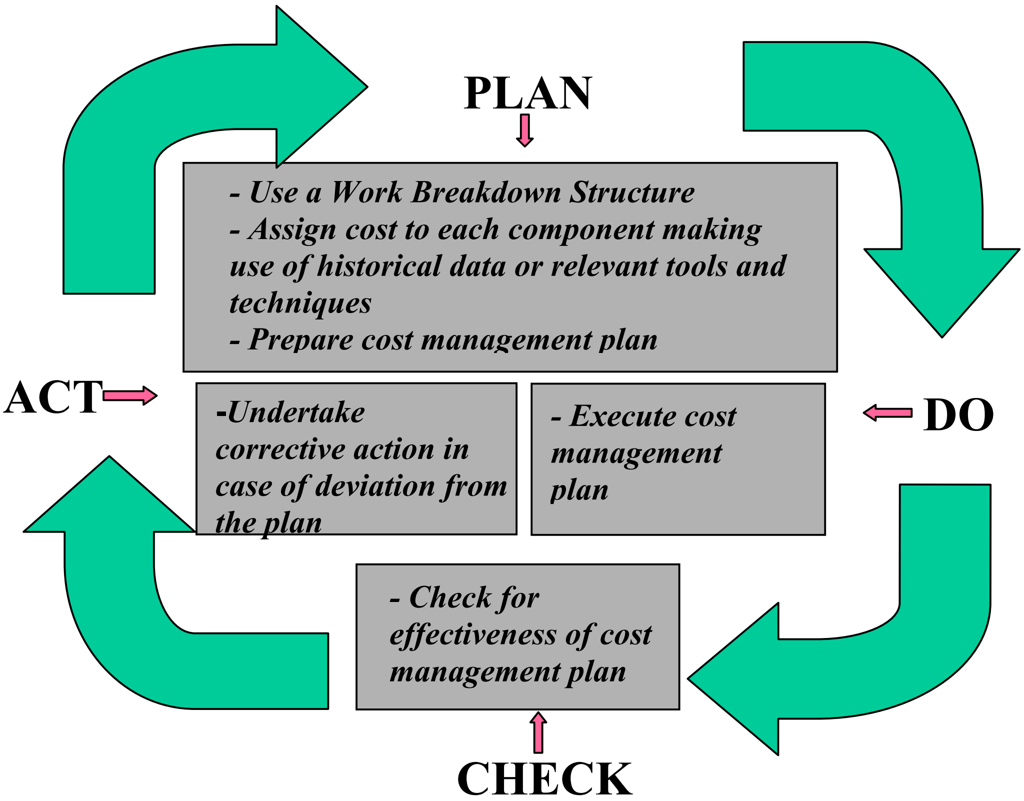


Figure 12.24: PDCA cycle for Cost management

Plan: Cost management

The planning for cost management was carried out from the person-month effort through analogy based on similar previous activities undertaken. In the case of programming, the work breakdown structure helped in subdividing a complex task into simpler tasks to which the object point analysis was applied in order to estimate the person efforts.

The total person-month effort was used to compute the cost of the project based on the rate applicable at the software development company.

Do: Cost management

The cost was tracked as the project progressed against its schedule. Microsoft Project was used in order to facilitate the process.

Check: Cost management

The actual cost was compared against the scheduled cost as the project progressed. Microsoft project helped in this process by keeping track of the remaining effort.

Act: Cost management

Any deviation from the planned estimates as far as delayed activities were concerned triggered concern and action on the part of the project manager to determine the reasons for any significant cost overrun. A situation of concern was noted during the planned acceptance test activity. Due to the reluctance of the users to conduct the acceptance test as planned, the activity was delayed. This situation was caused by an equipment relocation exercise to another computer room and the building was not completed in time. This problem was not foreseen as the information was not communicated to the software development organisation and was, therefore, not considered for risk management. If the problem was communicated, ESPM[3] would have been able to make provision for it by insisting on acceptance test to be carried out at the software development company's premises. The management of the user organisation would have had to take the commitment for a change in venue for the acceptance test.

Quality management

Quality management aims for:

- Customer satisfaction.
- Elimination of errors.
- Reduced operating costs.
- Motivation and commitment from employees.
- Profitability and competitiveness.

The transformation of the quality management inputs to the outputs is given in figure 12.25.

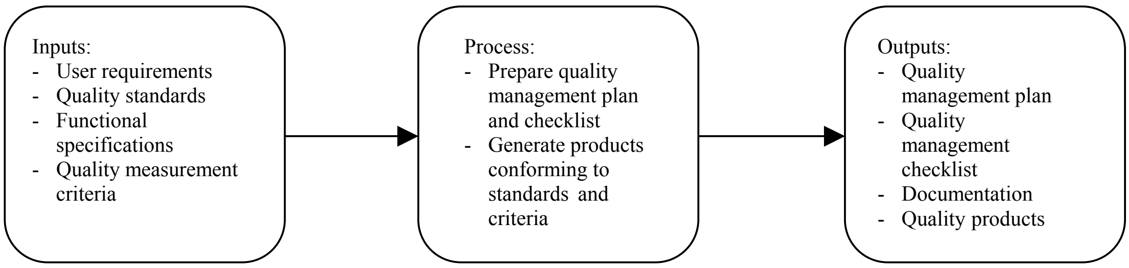


Figure 12.25: Quality management - Transformation of inputs into outputs

Quality management was mapped onto four process groups as per the PDCA cycle shown in figure 12.26.

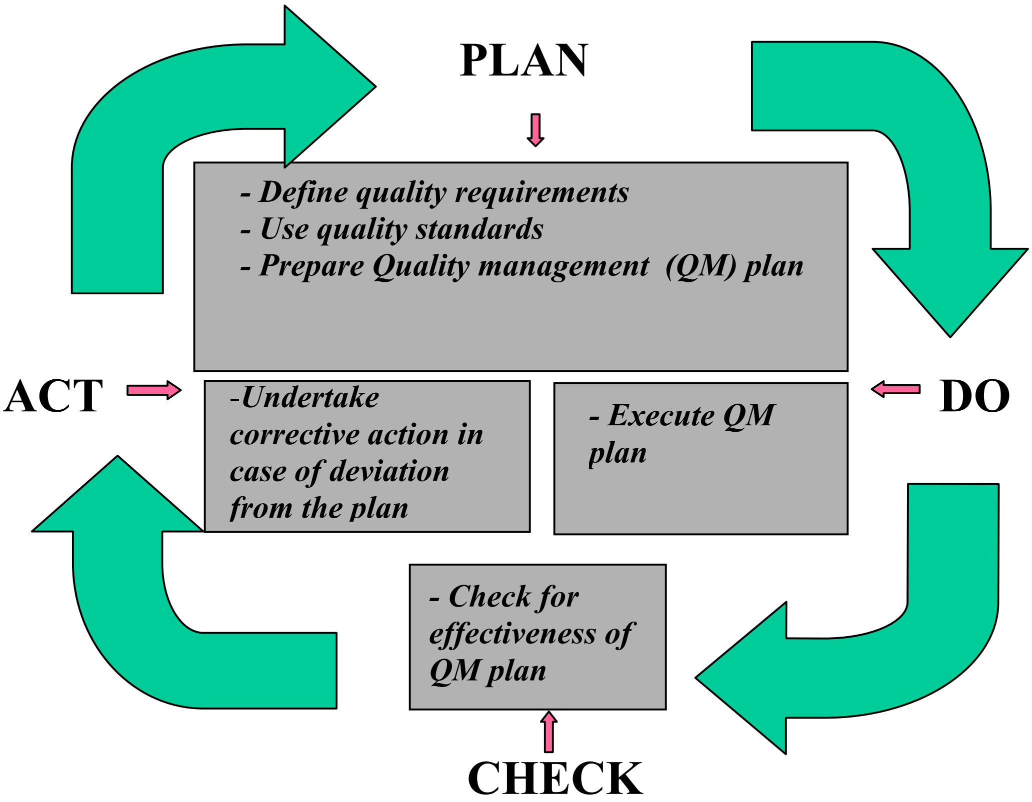


Figure 12.26: PDCA cycle for Quality management

Plan: Quality management

A quality plan was prepared and the quality targets for each deliverable were specified. For instance, the following products were identified to be subject to the quality control:

- analysis report.
- systems design.
- software product.
- documentation.

The analysis report and systems design were verified by the quality control team to ensure consistency and correctness. An ISO process (ISO 9001:2000) was applied to ensure consistency and to minimise errors. The software product was looked after by a program testing team and the quality control staff. In addition, the acceptance test activity was planned to allow the users to test the different functionalities of the software.

Do: Quality management

The plan was executed successfully until the acceptance test activity, which was disturbed by the unanticipated relocation exercise. With the quality management KPA, the quality of the product was considered as a key requirement for the ESPM^3.

Check: Quality management

During the "Do" process group, the data gathered at various checkpoints was verified against the predefined criteria to find out about any deviation. In ESPM^3, the quality of the software product is determined by the following criteria:

- Functionality.
- Usability.
- Reliability.
- Performance.
- Serviceability.

Acceptance test was found to be crucial for the project to be completed within the specified time and, therefore, according to the budget. The delay in the activity coupled with the loss of a key staff during the crucial period caused a delay in the delivery of the software product.

Act: Quality management

The delay in the acceptance test initiated action to be taken at the level of the finance section of the software development organisation. A request for payment, six months after the scheduled time prompted the management of department A to convene weekly meetings (instead of monthly meetings) in order to settle the problem. A full time staff member was assigned to help the users in conducting the acceptance test as well as correcting the bugs that were identified. The quality of the software as regards its reliability was questioned for the number of bugs that was identified. It was, therefore necessary to revisit the planning process group for all the KPAs of maturity level 2. It is to be noted that $ESPM^3$ provides a mechanism for several iterations through the use of the PDCA cycle until the product is of an acceptable quality.

12.3.2 At maturity Level 3

The three KPAs concerned here are Human Resource Management (HRM), Risk Management and Contract Management.

Human Resource Management

Human resource is the most important resource for a project (Nel et al., 2001) to be executed. The appropriate blend of skills is crucial in achieving the objectives of a project. Therefore, the team members need to be selected carefully and the project manager has to ensure that all of them are working in a conducive environment.

In $ESPM^3$, Human Resource Management can be carried out from the following inputs:

- Project team attributes, whereby the individual skills are identified and the appropriate blend of skills are taken on board.
- Organisational structure and constraints, whereby the relationships between the staff members are preserved according to the structure prevailing in the organisation and its constraints.
- Employee relations constraints, whereby care has to be taken so that no employee's rights are violated.

The inputs of the process generate outputs in the form of:

- HRM plan detailing the human resources assigned to each activity.
- Documentation on the human resource management plan.

The transformation of the inputs to the outputs is shown in figure 12.27.

Figure 12.27: HRM - Transformation of inputs into outputs

Human Resource Management was mapped onto four process groups as per the PDCA cycle shown in figure 12.28.

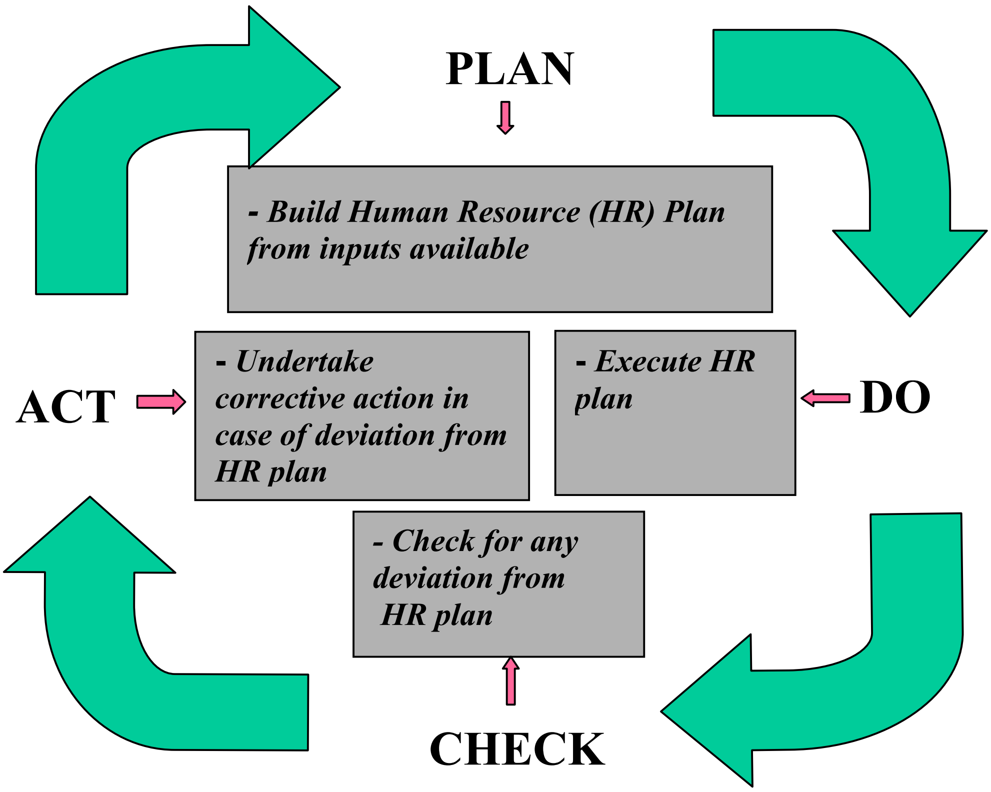


Figure 12.28: PDCA cycle for Human Resource management

Plan: Human Resource Management

The number of resources assigned to the different activities of the project was determined through the planning process group. Effective human resource management is important as it provides for:

- more effective and efficient use of human resources.
- greater employee satisfaction.
- reduced conflicts at work.

Microsoft Project was used to assign resources to the different activities of the project. At a glance, the different resources consumed could be identified from the Gantt chart. Additional information concerning the basic rate per hour and the overtime rate per hour were also available.

Another tool used to enhance the visibility of resources assigned to each activity was the responsibility assignment matrix. The table produced for the purpose of this project is shown in table 12.9. The phase assigned to each staff member is marked with a cross (X).

Table 12.9: Responsibility Assignment Matrix

Phases \ Staff	A	B	C	D	E
Feasibility	X				
Analysis		X			
Design		X	X		
Coding			X	X	
Testing					X
Training		X	X	X	
Implementation		X	X	X	

The planning process group also took care of the planning for the selection of project team members and ensured a suitable employee relationship. ESPM[3] helped in the planning phase for HR given that it comprised a KPA for HRM.

Do: Human Resource Management

This process group involved the execution of the human resource management plan through the selection of an appropriate team and maintaining an appropriate blend of skills as well as favouring a conducive environment while the project is in progress.

Check: Human Resource Management

The project manager played an important role in monitoring the tasks performed by each team member and ensured that he/she was working according to the time schedule prepared. During this process, conflicts were identified and eliminated so as not to affect the smooth execution of the project.

At the end of November 2007, it was found that the project could not be completed due to a change in technology with respect to the database management system and the construction of a new computer room. Even with an extended schedule till the end of June 2008, the project was not completed due to a lack of user commitment in conducting the acceptance tests successfully. In addition, the programmer left the organisation at the crucial period of acceptance testing. A new time schedule had to be prepared in June 2008 in order to complete the project by September 2008.

Act: Human Resource Management

The project manager had to assign the remaining activities to a new programmer. The replacement of the programmer was done with due care in order to minimise the delay. This problem was appropriately handled by the risk management process group. Therefore, $ESPM^3$ was found to be suitable to deal with the impact brought by a key staff leaving the organisation.

Risk Management

Risk management is concerned with the mitigation or elimination of unforeseen events that may pose a threat to the successful completion of a project. It is achieved by avoiding these risks or by drawing up risk management plans for dealing with them. The project manager was responsible for managing the risks that could cause the late delivery of a project, a budget overrun or a poor quality software to be developed.

Inputs are identified and the processes to transform these into outputs are defined. The following inputs were considered:

- List of potential risks.
- Historical data.

The risk management inputs were processed and outputs were generated in the form of:

- Prioritised risks according to their likelihood and impact.
- Risk management plan.
- Documentation.

The transformation of the inputs to the outputs is shown in figure 12.29.

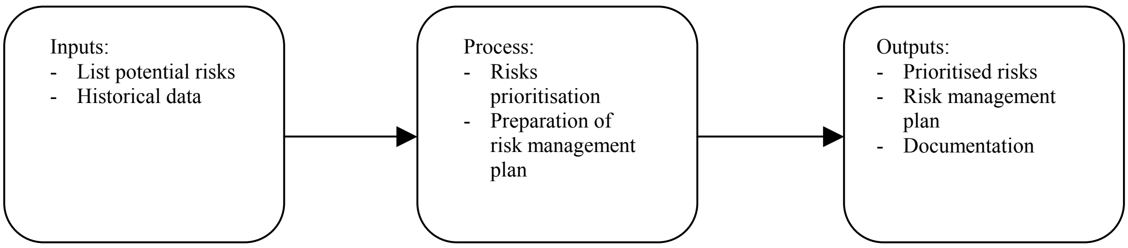


Figure 12.29: Risk management - Transformation of inputs into outputs

Risk management was mapped onto four process groups as per the PDAC cycle shown in figure 12.30.

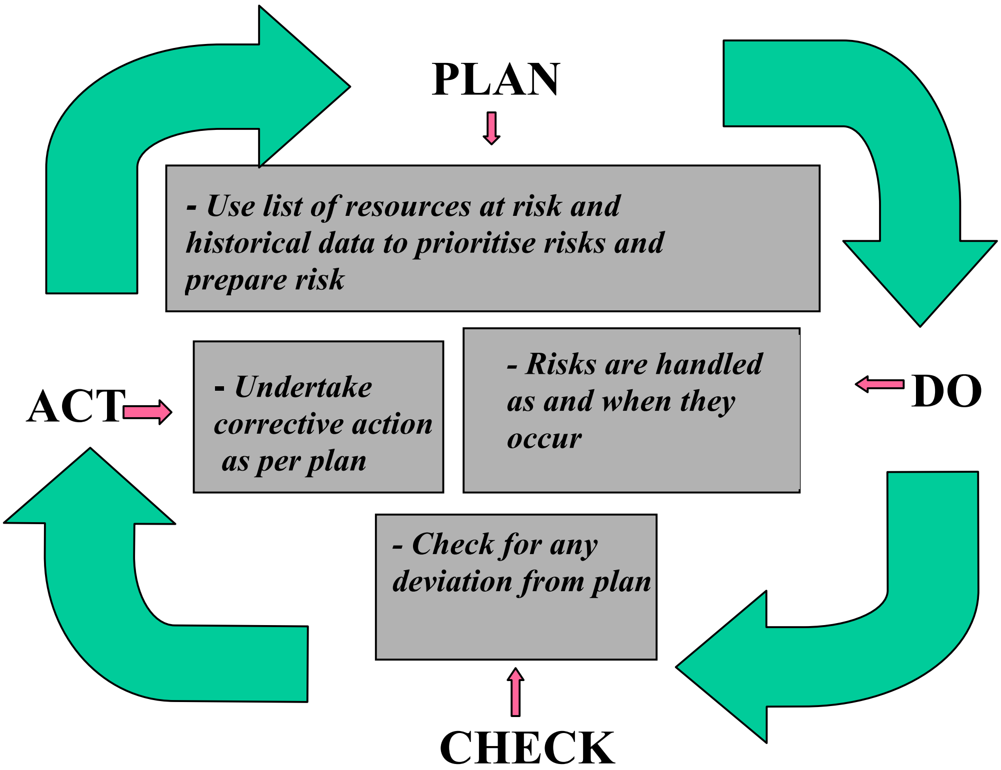


Figure 12.30: PDCA cycle for Risk management

Plan: Risk Management

Those risks that posed a threat to the project were identified, evaluated and prioritised. The potential hazards affecting the project resources were listed. Values between 1 and 5 were assigned to the risks identified with the following denotations:

1 = No impact
2 = Low impact
3 = Moderate impact
4 = High impact
5 = Very high impact

Similarly, the likelihood for the occurrence of an identified risk hazard was also assigned a value from 1 to 5.

The risk analysis table is shown in table 12.10.

Table 12.10: Risk analysis table

Risk	Likelihood	Impact	Risk exposure
Changes to specifications	1	4	4
A key staff being ill or leaving the job during critical activities	1	3	3
Analysis taking more time than planned	2	4	8
Design taking more time than planned	2	4	8
Coding taking more time than planned	2	4	8
Users not satisfied with the quality of the software in terms of the functionality	1	5	5

Risk exposure is the product of the likelihood of a risk hazard occurring and its impact on the successful delivery of the project.

The risks were then prioritised in terms of their risk exposure value. A new table (table 12.11), sorted in descending values of risk exposure, was generated.

Table 12.11: Risk exposure table

Risk	Likelihood	Impact	Risk exposure
Analysis taking more time than planned	2	4	8
Design taking more time than planned	2	4	8
Coding taking more time than planned	2	4	8
Users not satisfied with the quality of the software in terms of functionality	1	5	5
Changes to specifications	1	4	4
A key staff being ill or leaving the job during critical activities	1	3	3

The project manager gave special attention to the high risk exposure values.

A risk management plan, elaborating on the means to deal with each risk hazard, was also prepared as shown in table 12.12 and the plan allowed the project manager to adopt a proactive measure to deal with risks instead of adopting a crisis management measure.

Table 12.12: Risk management plan

Risk	**Likelihood**	**Impact**	**Risk exposure**	**Method to deal with the risk hazard**
Analysis taking more time than planned	2	4	8	Overtime hours to be considered.
Design taking more time than planned	2	4	8	Overtime hours to be considered.
Coding taking more time than planned	2	4	8	Adding more people on the activity. Given that the software was to be developed in modules and each module had its own specifications prepared by the systems analysts, it would possibly be easier for any programmer of the software development organisation to be assigned certain tasks.
Users not satisfied with the quality of the software in terms of functionality	1	5	5	Testing and quality control to be effected by experienced staff. The prototype need to address functionality issues.
Changes to specifications	1	4	4	The user has to bear the additional cost according to the contract.
A key staff being ill or leaving the job during critical activities	1	3	3	Replacement staff was identified.

Do: Risk management

Risk management was carried out using the risk management plan developed in the “Plan” process group of ESPM[3]. The risk management plan helped in addressing many risk hazards. However, lack of user commitment for the acceptance testing and a lack of management support caused a delay in the project. In addition, the construction of a new computer room was not foreseen. A delay was also introduced during the programming activity due to a change in the version of the database management system for which the programmer did not have previous experience. The delay was for a good cause as the user could benefit from a new version of the software making use of better technology.

Check: Risk management

This process group allowed the project manager to monitor for the occurrence of any risk hazards. With the help of the risk management plan, the occurrence of each potential risk hazard was identified as soon as it occurred for subsequent action.

Act: Risk management

The loss of a key staff member during the acceptance testing activity was adequately handled by his immediate replacement. The availability of documentation and the proper handling over process ensured negligible delay.

Contract management

A contract, being a formal agreement between two parties, obligates one party (the software development organisation) to perform a service and the other party (the client) obligates itself to do something in return, usually in the form of a payment. In the case of the project, it was a direct procurement exercise as it was a review of an existing system whereby the supplier of the existing systems was requested to carry out the review exercise.

In this project, contract management made use of the following inputs:

- Requirements prepared by client.
- Proposal from the contractor.
- Clauses in the contract and deliverables.

The output generated was:

- Successful contract execution.
- Acceptance of deliverables by the client.

The transformation of the contract management inputs to the outputs is shown in figure 12.31.

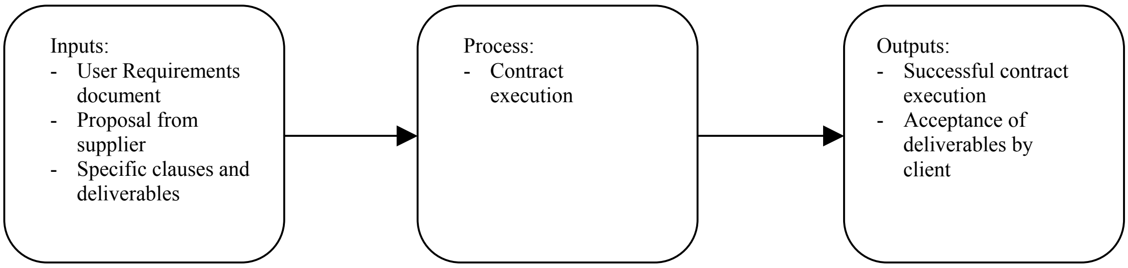


Figure 12.31: Contract management - Transformation of inputs into outputs

The project manager was aware of the implications of the clauses in the contract and he ensured as far as possible for compliance unless there was mutual agreement of performing the tasks differently.

The mapping of contract management onto the four process groups as per the PDCA cycle shown in figure 12.32.

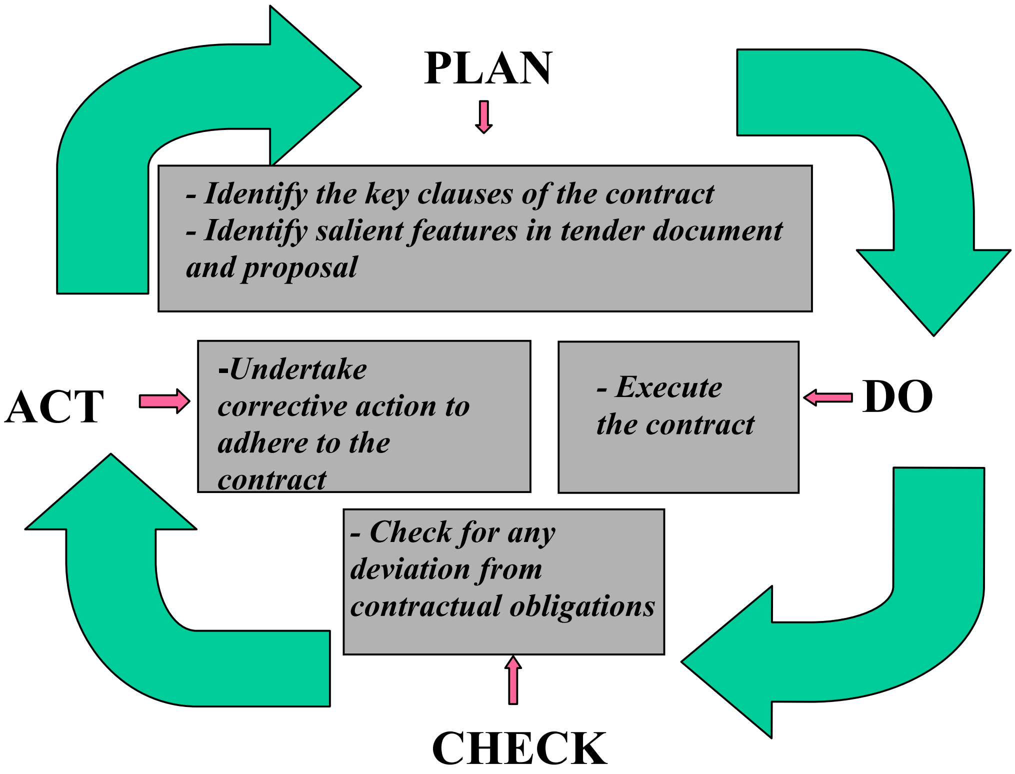


Figure 12.32: PDCA cycle for Contract management

Plan: Contract management

Since preparing a software contract is an activity to be performed by a legal person, standard contracts for the organisation were prepared by a legal advisor. The contract was used with minor modifications. Upon signing of the contract, the project was started. In the contract, the parties concerned, the work to be carried out, the terms of payment, and the deliverables to be submitted at a specified period in time were specified in addition to other clauses like the extension of the contract in case of act of God, cancellation of the contract and penalties to be applied in the case of non respect of the contractual agreements.

Work around solutions were foreseen in the case of any problem. In the case of this project, it was planned for acceptance testing and training to be carried out at the premises of the software development organisation.

Do: Contract management

Upon signing of the contract, a percentage of the contract value was claimed from the client in order to start the project. All deliverables that had to be provided were noted together with their dates of submission against the plan.

A list of important issues in the contract was prepared by the project manager to ensure that all clauses forming part of the contract were strictly adhered to during execution.

Check: Contract management

The deliverables were checked regularly against the plan to facilitate their timely delivery. In the case of this project, it was found that the user did not perform acceptance testing in time since the software was not available for the test according to the plan. In addition, construction works (for a new computer room) caused further delays and when the user decided to carry out the acceptance test, they found several bugs. Given that there were delays caused by both parties, it was mutually agreed upon not to apply the penalty clauses.

Act: Contract management

Following the identification of several bugs in the software during acceptance testing, the client decided not to release the last payment as per the plan for payment. A revised Gantt chart was prepared and a new plan for payment was accordingly agreed upon between the software development organisation and the client.

12.3.3 Continuous Process Improvement group of KPAs

The five KPAs concerned here are Soft Skills Management, Change Management, Software Specific focus, Environmental Management and Integration Management.

Soft Skills Management

Soft skills management, concerned with the management of and working with people, were applied to this project. Soft skills management was carried out through the following inputs that were transformed into outputs:

- List of soft skills.
- Soft skills remedial action knowledge base.
- Historical data.

The output generated was:

- Satisfied team members.
- High performance and productivity.
- Lessons learnt to be used for future soft skills management.

The transformation of the inputs to the outputs is shown in figure 12.33.

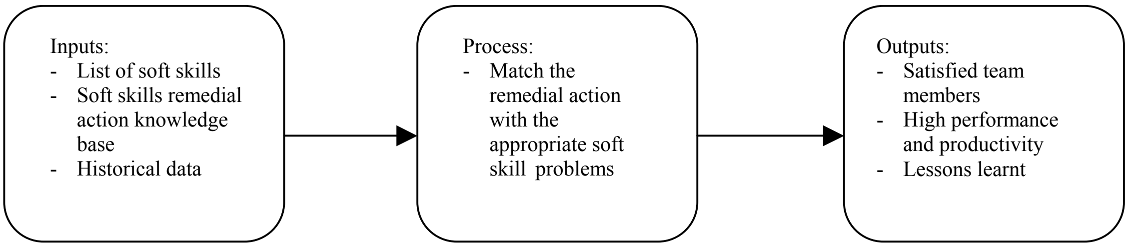


Figure 12.33: Soft skills - Transformation of inputs into outputs

The PDCA cycle for Soft Skills Management

Soft skills management was mapped onto the four process groups as per the PDCA cycle and as per requirements of ISO 9001:2000 standard (Mauritian standard, 2001).

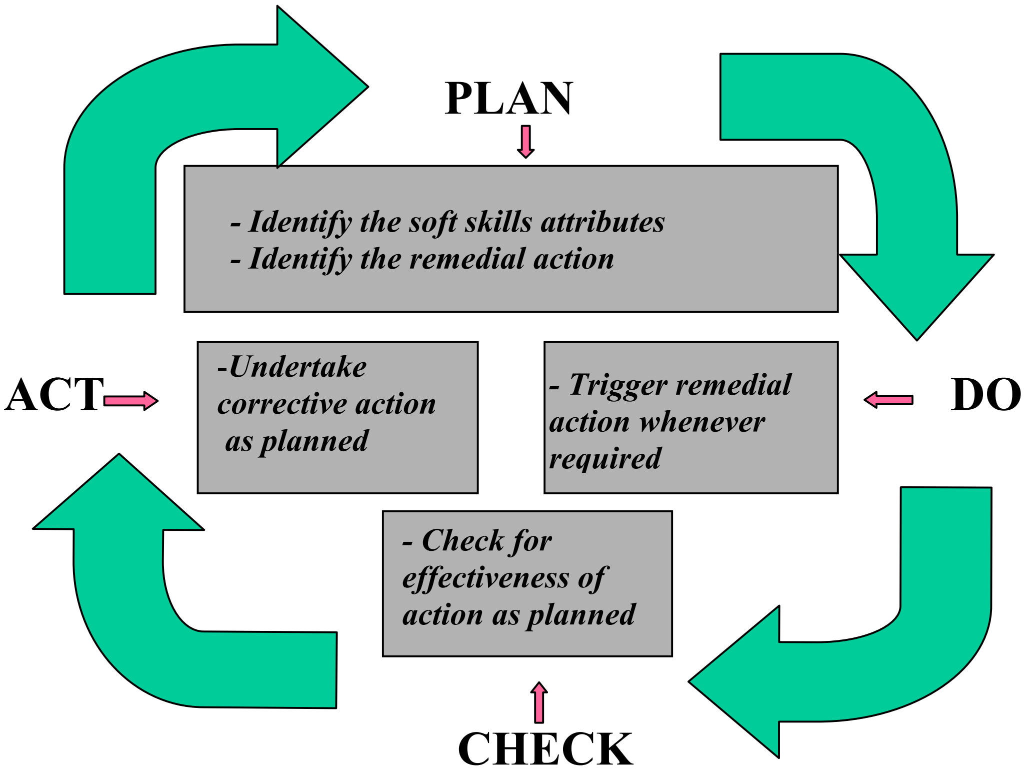


Figure 12.34: PDCA cycle for soft skills management

The soft skills considered were Communication skills, Team building, Creativity, Leadership, Organisational culture dissemination, Stress management, Trustworthiness and Conflict management.

Plan: Soft Skills management

The planning process group is intended to be the stage for listing all the possible remedial actions to be taken to deal with certain shortcomings arising from management of staff. Table 12.13 was generated to implement soft skills. Initially the rightmost column is blank and it is populated during the project.

Table 12.13: Soft skills management plan

Problem requiring soft skills management	Soft skills attribute	Possible remedial actions	Appropriate action for the project
Inadequate Communication	Communication skills	Honest discussion	
		Listen to person concerned	
		Negotiate	
		Facilitate discussion	
Inappropriate blend of skills	Team building	Exchanging team members between teams	
		Participation of a member from another team to provide services on a part time basis	
Lack of user-friendliness of software	Creativity	Feedback from users	
		Use of standards developed within the organisation	
De-motivated staff	Leadership	Consensus building through listening and understanding the reasons for de-motivation	
		Use of appropriate leadership style	
		Provision of appropriate incentives and rewards	
Opposing organisational culture	Organisational culture dissemination	Holding discussion to facilitate commitment	
		Provide information on the need for the prevailing culture for efficiency in the organisation	
Excessive stress originating from need to ensure delivery of project within time, cost and quality constraints	Stress management	Regular feedback on status of all aspects of a project and providing any corrective action	
		Use of appropriate communication channel for reporting of problems pertaining to delivery of project so that corrective action is taken	
Lack of trust	Trustworthiness	Discussion, listening and understanding	
		Consensus building to enhance trust	
Conflict between team members	Conflict management	Mediation	
		Arbitration	
		Control	
		Acceptance	
		Elimination	

Do: Soft Skills management

During the execution of the project, the relevant soft skills were identified and the appropriate actions were taken as per the table prepared during the planning phase. In table 12.14, the relevant action was marked with a tick.

Table 12.14: Execution of soft skills

Problem requiring soft skills management	Soft skills attribute	Possible remedial actions	Appropriate action for the project
Inadequate Communication	Communication skills	Honest discussion	
		Listen to person concerned	
		Negotiate	
		Facilitate discussion	√ The Project Manager intervened during the execution of the project in order for training to be carried out at the contractor's site as the client's site was not ready during the acceptance testing phase. Unfortunately, the unanticipated Civil Works undertaken by the client caused major delay during the acceptance testing phase.
Inappropriate blend of skills	Team building	Swapping team members between teams	
		Participation of a member from another team to provide services on a part time basis	
Lack of user-friendliness of software	Creativity	Feedback from users	
		Use of standards developed within the organisation	
De-motivated staff	Leadership	Consensus building through listening and understanding the reasons for de-motivation	
		Use of appropriate leadership style	
		Provision of appropriate incentives and rewards	

Problem requiring soft skills management	Soft skills attribute	Possible remedial actions	Appropriate action for the project
Opposing organisational culture	Organisational culture dissemination	Holding discussion to facilitate commitment	
		Provide information on the need for the prevailing culture for efficiency in the organisation	√ The software development team was already familiar with the company's culture and they were conversant with a methodology for software development.
Excessive stress originating from need to ensure delivery of project within time, cost and quality constraints	Stress management	Regular feedback on status of all aspects of a project and providing any corrective action	√ As from the acceptance testing phase, the stress started to build up and there were several interventions required by the Project Manager to get the project completed. Unfortunately, there was difficult to have management support as well as user support. Following the claim for payment, the management of the client organisation decided to discuss so that the project could meet its objectives.
		Use of appropriate communication channel for reporting of problems pertaining to delivery of project so that corrective action is taken	
Lack of trust	Trustworthiness	Discussion, listening and understanding	
		Consensus building to enhance trust	
Conflict between team members	Conflict management	Mediation	
		Arbitration	
		Control	
		Acceptance	
		Elimination	

Even though a small team undertook the development of the project, the minimum soft skills attributes were still found to be necessary. However, the good communication skills among the software development team ensured conducive working relationships. It could be noted that even though one staff member left prior to the completion of the project, handing over was adequately done so as not to cause the project to fail and allow the software development company to be in an embarrassing situation vis-à-vis the client.

Check: Soft Skills management

The Project Manager ensured good communications among the development team. Right at the start of the project, the Project Manager also ensured that an appropriate blend of skills was available. The intervention of the Project Manager was necessary at various stages so as to reduce delays or prevent failure of the project. Even when discussions with the management of the client's organisation were not successful, the need to claim for payment was necessary in order to acquire the necessary management support and revise the remaining activities of the project.

Act: Soft Skills management

When it was noted that there was no progress made towards the completion of the project, it was necessary to acquire management support. Weekly management meetings (instead of monthly meetings) were requested by the project manager and any issues were dealt with in the shortest time possible. The tasks to be completed prior to the next meeting were assigned to the officers concerned.

The PDCA cycle was continuously executed till the end of the project. For instance, it was required to complete a cycle in between every two meetings.

The details on change management, software specific focus, environmental management and integration management are found at appendix E.

Finally the project was completed by end of September 2008, that is one year later. The main reasons were the unforeseen circumstance regarding the construction of a building comprising a new computer room as well as user resistance and commitment to test the software during the acceptance testing phase.

ESPM^3 provided a suitable environment for the execution of the project, which could have been completed on time, within budget and with the quality required. Adequate support of staff from the client's organisation was required for this project. Acceptance testing could have been carried out at the software development organisation and the software product would have been delivered on schedule in the event that the users were committed.

12.4 Conclusion

Two case studies were presented in this chapter. It should be noted that the project dealing with the implementation of a *job information system* was implemented successfully using ESPM^3. All the KPAs of maturity levels 2 and 3 as well as the continuous process improvement group were applied. The normal course of action would have been to test the KPAs of maturity level 2 together with the continuous process improvement group in the first instance and assess the maturity level of the organisation. After the KPAs had reached maturity level 2, then the maturity level 3 KPAs would have been applied. The application of all KPAs requires additional resources and effort and there is a need to ensure that the organisation has at least successfully attained maturity level 2 in the first instance.

The *second case study* dealt with the implementation of a recruitment system using ESPM^3. Again all the KPAs of the model were applied. Although a significant delay was noted in the implementation of the project, this was not attributed to a weakness in the model. It was rather due to a lack of support on the users' side although much effort was made by the project manager and the project team to rectify this.

It is, therefore, concluded that ESPM^3 could be applied to ensure successful project delivery through an improvement in the maturity level of organisations. As mentioned previously, an improvement in the maturity level is related to the success of project delivery.

In the next chapter, the conclusion of this thesis is presented along with an analysis of the research and the future work to be undertaken.

Part 4

Conclusion and Future Work

Chapter 13

Analysis, Conclusion and Future Work

Table of contents

13.0 Chapter layout

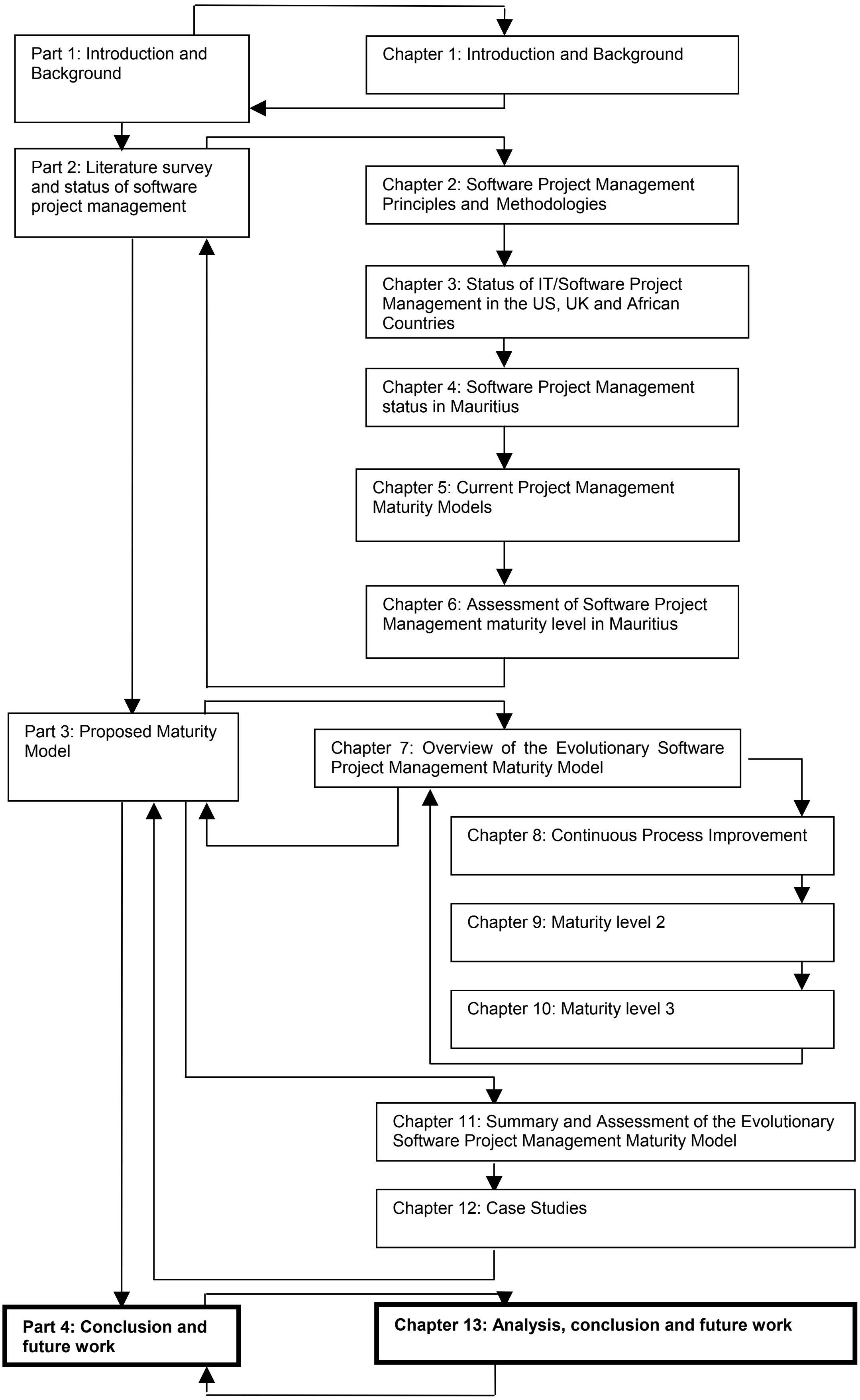


Figure 13.1: Chapter 13 in context within the overall thesis

13.1 Introduction

This chapter comprises a conclusion of the thesis. As there were problems pertaining to software project management in developing countries, it was speculated that a new software project management methodology/model/standard could help in improving the success rate of software projects. Therefore, a main objective was set for the development of a new methodology/model/standard for developing countries so as to take into consideration the nature of the problems encountered. The main objective was disseminated into two secondary objectives which were highlighted in chapter 1. Shortcomings in the available project management methodologies/models/standards as well as the tools and techniques that can be used to improve software project management were also identified. Therefore, the achievement of these objectives and their deliverables are discussed in this chapter.

In the last section of this chapter, an insight into the future work that can follow from the development of $ESPM^3$ is discussed.

13.1.1 Aim, objectives and structure of this chapter

In this chapter, the summary, conclusion and future work of the thesis are presented.

The remainder of the chapter is structured as follows:

Section 13.2 – In section 13.2, the contribution of this thesis is presented on the basis of the objectives set out in chapter 1. The extent to which the objectives have been met is discussed.

Section 13.3 – In this section, an analysis of $ESPM^3$ is presented by taking as example its application to two case studies.

Section 13.4 – The strengths of ESPM^3 are presented in this section.

Section 13.5 – The weaknesses of ESPM^3 are considered in this section.

Section 13.6 – In section 13.6, an indication of future work that may be conducted for further improvement of the model is discussed.

In order to show the applicability of the ESPM^3, two projects were chosen and all the KPAs were applied. One project was successful as it was found to be executed within the initially specified time, budget and quality. The second project did not meet the expectation as regard the duration aspect. It could, however, be noted that the failure to meet the objectives of the project was attributed to the lack of executive support as well as unavailability of the client's support during the testing phase. Furthermore, an unanticipated relocation of the computer facilities at the client's site caused major disruption in the schedule of work. Therefore, it could be concluded that ESPM^3 was suitable for software projects of political importance, having a social impact and developed in an environment different from that of developed countries.

13.2 Contribution of this thesis

In chapter 1, the main problem area regarding the weaknesses of software project management approaches of Western origin in African developing countries was pointed out. In chapter 2, the existing project management methodologies/models/standards were explored in order to find out about their strengths and weaknesses/shortcomings for developing countries. The available project management methodologies/models/standards were found to lack the feature to deal with the cultural, social, economic and political aspects in connection with software projects. The main problem was the specificities of the situation in developing countries as mentioned by Muriithi and Crawford (2003) as well as by researchers as explained in chapter 3. The need for an indigenous approach was even pointed out.

Therefore, specific problems had to be addressed as it was found that project management approaches of Western origin were not universally applicable as they had certain weaknesses. They assumed economic, social, cultural and political rationalities. It was, therefore, speculated that a software project management methodology/model/standard that could address weaknesses would assist in the successful delivery of software projects in developing countries.

An assessment of the state of IT and Software Project Management in developed countries like the US and the UK as well as developing African countries was carried out. The gap regarding IT and Software Project Management between developing and developed countries was highlighted by considering Hofstede's dimensions. The results again showed that these dimensions had an influence on project management in general.

In chapter 3, the unsuitability of Project Management approaches of Western origin were confirmed by Rondinelli (1976), Stuckenbruck and Zomorrodian (1987) , Ronan (1993). The findings are still valid as confirmed by the research conducted by Muriithi and Crawford (2003).

In addition, Ronan (1993) emphasised on the characteristics of the African manager and a remarkable difference could be noted with their counterparts in the developed world. African managers were known to have a high need for relationships and status and a low need for the achievement of objectives. The failure of Western project management models was also pointed out.

In chapter 4 of this thesis, the status of Software Project Management in Mauritius was investigated through a survey. The lack of the use of formal software project management methodologies/models/standards and project management tools were noted as software companies were not achieving the expected results. There was also an important lack of skilled human resources in the country, which contributed to the deviation in achieving the project outcomes.

The applicability of a maturity model was also explored. A literature survey of existing project management maturity models was carried out. Their strengths and weaknesses were identified. Again, the lack of the necessary features to cope with the political, social and cultural aspects was noted. The general nature of most models rather than the software specific one, the associated voluminous documentation and the bureaucratic procedure involved were apparent.
The choice for a maturity based model was made following the correlation observed between project success and project management maturity level of organisations by Schiltz (2003) and Sonnekus and Labuschagne (2004). The maturity level of software development organisations in Mauritius was assessed in a manner similar to that conducted in South Africa.

In this thesis, the identification of the KPAs for the maturity model for developing countries were made following an interview. The grouping of the KPAs into three maturity levels was carried out resulting in the development of the $ESPM^3$.

The new model ($ESPM^3$) was developed to take care of the political, social, cultural and economic situation prevalent in developing countries. Soft skills were also considered by including a Soft skills management KPA. The model is a customisable one, where KPAs may be omitted in the event that they are irrelevant. In the case of some Mauritian organisations, it was found that the Cost management KPA was not relevant for in-house development. KPAs at higher levels may also be incorporated at lower levels in case there is such a need. $ESPM^3$ is also a software specific model, which takes into consideration the strengths and weaknesses associated with software projects. It also has a simple assessment method to assess the maturity level of an organisation which was described in chapter 11.

The case of Mauritius as an instance of developing countries was considered for the research. An objective (**primary objective**) was set in order to develop such a methodology/model/standard to enhance the success rate of software projects. Secondary objectives were also explored together with their deliverables as follows:

Objective 1: To **identify shortcomings** in available project management methodologies/models/standards and to develop a software project management methodology/model/standard for Mauritius (**deliverable 1**).

Objective 2: To **identify tools and techniques** (**deliverable 2**) to be used with the software project management methodology/model/standard.

The extent to which these objectives have been met are discussed below.

(i) Objective 1 and deliverable 1 – The shortcomings in currently available methodologies/models/standards were pointed out. The problems in developing countries, like Mauritius, required a different approach in order to manage software projects. Over several years of research, Rondinelli (1976), Stuckenbruck and Zomorrodian (1987), Ronan (1993) and Muriithi and Crawford (2003) emphasised on the need for a different approach for managing software in general in African developing countries. Following researches conducted by Sukhoo et al. (2004b and 2005a), a software project management model called the evolutionary software project management maturity model (deliverable 1) was put forward and tested in the Mauritian environment.

(ii) Objective 2 – Tools and techniques like the Gantt chart, Critical Path Method, estimation by analogy, object point analysis, software tools like Microsoft project or any similar software (deliverable 2) were recommended for use to assist project managers and team members to achieve the objectives of their projects.

The primary objective was met through the development of an Evolutionary Software Project Management Maturity Model (ESPM^3) for developing countries. As explained above, the model was developed through surveys and interviews conducted in Mauritius. On the strength of information gathered, ESPM^3 was built with three maturity levels and it took into consideration the needs and constraints of the country. As mentioned in chapter 1, Mauritius was chosen as an instance of developing countries. The problems encountered with respect to project management were similar to those of developing countries as pointed out by Muriithi and Crawford (2003).

As such, this thesis contributed towards an exploration of the situation related to software project management in developing countries. It was found that project management approaches of Western origin were not universally applicable. It was shown that an indigenous software project management approach was necessary for developing countries. In this context, a new Software Project Management Maturity Model ($ESPM^3$) was developed to take care of the specificities of the problems in developing countries. In contrast to many of the existing models, $ESPM^3$ is a software specific model that considers soft skills, is customisable and avoids the bureaucratic procedures associated with most methodologies/models/standards. It does not possess voluminous documentation, thus helping project managers to adopt it with the minimum effort. $ESPM^3$ also provides recommendations on project management tools to use with the software projects.

13.3 Analysis

In this thesis, we conducted two studies to assess the status of software project management in Mauritius as an instance of developing countries (Sukhoo et al., 2004b; Sukhoo et al., 2005a). A literature study was also undertaken to determine the status and problems being encountered in the US, UK and developing countries. These studies showed that there were difficulties in improving the success rate of software projects.

In order to show the applicability of the $ESPM^3$, two projects were chosen and all the KPAs were applied. One project was successful as it was executed within the initially specified time, budget and quality. The second project did not meet the expectation as regard to the duration aspect. It could, however, be noted that the failure to meet the objectives of the project was attributed to the lack of executive support as well as an unavailability of the users' support during the testing phase. Furthermore, an unanticipated relocation of the computer facilities at the users' site caused major disruption in the schedule of work. Therefore, it could be concluded that $ESPM^3$ was suitable for software projects of a political importance, having a social impact and developed in an environment different from that of developed countries.

The successful application of $ESPM^3$ and recommended tools and techniques showed the suitability of the model for Mauritius. It is believed that the adoption of $ESPM^3$ will help project managers to manage their projects better and ensure the timely delivery of their projects within budget and at the level of quality that will satisfy the users.

However, $ESPM^3$ was found to have some weaknesses along with its strengths as explained in the following subsections.

13.4 Strengths of $ESPM^3$

The following strengths may be noted for $ESPM^3$:

- $ESPM^3$ is a software specific model as compared to most of the project management methodologies/models/standards studied in chapters 2 and 5 of this thesis. Therefore, it is specifically geared towards the specificities of software projects. Software projects were found to have certain inherent strengths and weaknesses (Sukhoo et al., 2004a). By taking advantage of these strengths and adopting measures to mitigate the weaknesses, the use of $ESPM^3$ may improve the success of software projects by reducing the delivery time, enhancing the quality of the product and also reducing the cost of development.
- None of the project management maturity models studied in chapter 5 took care of soft skills which were found to be important in managing software projects. $ESPM^3$ has a key process area dealing with soft skills. Soft skills enhance communication, team building, creativity, leadership, organisational culture dissemination, stress management, trust and

conflict management. As pointed out in chapter 8, according to Belzer (2004), few projects fail because of hard skills. Instead, projects fail due to a lack of soft skills. The soft skills required for managing software projects in the Mauritian context were presented by Sukhoo et al. (2005b).

- Voluminous documentation was associated with most of the methodologies/models/standards studied in chapters 2 and 5. ESPM^3 was designed to avoid excessive documentation so as to encourage its adoption by organisations. The lack of human resources in developing countries means that project managers will not have sufficient time to go through voluminous documentation in order to start with the use of the model.
- The bureaucratic nature of most methodologies/models/standards have prevented Mauritian software development organisations from adopting them. In ESPM^3, lengthy and inflexible procedures have been avoided. The model can also be customised to the needs of the organisation. Some KPAs may even be ignored in case they are not applicable. Other higher maturity level KPAs may also be applied at lower maturity levels.
- ESPM^3 includes one key process area for the social, economic, cultural and political factors. These factors had a major impact on certain projects. For example, the projects in Mauritius were given priority in case they had political, social, economic and cultural benefits and sometimes at the expense of other important projects.
- Time management is performed through the use of the recommended project management tools and techniques (like the Gantt chart, WBS and software project management tools). The PDCA cycle enabled an iterative process to be followed until the project is completed. The use of a project management software, like Primavera and Microsoft Project, is also recommended by ESPM^3.
- The cost of the project is managed by mapping the cost management KPA onto the PDCA cycle. The planned cost is calculated by the tools and techniques (like WBS, object point estimation, estimation by analogy and software tools such as Microsoft Project) recommended with ESPM^3.

- ISO 9001:2000 was chosen as the quality standard for Mauritius, although one company used IEEE standard (see case study 1 in chapter 12). Companies in various developing countries are free to choose the quality standard that they desire.
- ESPM^3 also included the KPA for HRM. This KPA was found to be important for managing software projects (Sukhoo et al., 2007) as it allowed the organisation and the project manager to choose the appropriate blend of skills for the development of a software product. This may also facilitate the communication between the project manager and the team members as well as maintain trust among the staff members. PRINCE 2® excluded HRM as it is considered to be separate from the methodology. However, in the case of Mauritius, HRM was found to be necessary to be dealt with at both the organisational level as well as the project team level.
- Risk management is handled in ESPM^3 by a KPA mapped onto the PDCA cycle so as to reduce risks through an iterative process. A relatively well known procedure was recommended for the identification, evaluation and prioritisation of risks.
- Contract management is also included in ESPM^3. According to the survey carried out by Sukhoo et al. (2007), the inclusion of a contract management KPA was found to be important. Therefore, it was included and it allowed project managers to be careful about compliance to the clauses of the contract. For instance, a project manager needs to be constantly aware of any penalty applicable for the late delivery of a software product. Cost overrun may occur in the event that the project manager does not pay attention to the agreed date of delivery of a software product.
- ESPM^3 can facilitate the exposure of the staff members to forthcoming technologies through training. The use of Beta versions are encouraged for training so that the development staff member is aware of the technology prior to the release of the new version of the DBMS.
- Given the features of ESPM^3 (with smaller documentation, flexibiliy for customisation, ability to cope with the political, social, cultural and

economic situation in the country as well as the ability to deal with soft skills and the software specific nature) as compared to other methodologies/models/standards, it will be easier to adopt for developing countries like Mauritius in order to develop the sector into a pillar of the economy. Software development companies can easily assess their maturity levels at any stage in order to decide on further improvement needed. Since most companies in Mauritius have not adopted a software project management methodology/model/standard of Western origin, it is expected that $ESPM^3$ will be accepted in the country as well as in other developing countries.

13.5 Weaknesses of $ESPM^3$

The following weaknesses were also noted during the implementation of the model:

- As mentioned previously, $ESPM^3$ may be easily customised in the event that some companies have a different settings (for example, it was mentioned that in some organisations in Mauritius, cost management was not applicable). This may also mean that there may be some reluctance to use a model that may require some extra work to be performed in order to be used exclusively by these companies.
- The relatively large number of KPAs in $ESPM^3$ may increase the load of work of the project manager. We are of the opinion that the development of an automated software tool may assist project managers to manage software projects efficiently.

13.6 Lessons learnt from this research

This research has shown that project management concepts and approaches in the developing world are different from that of developed countries. Developing countries are affected by a lack of funds and skilled human resources. In addition, the political, social, cultural and economic situations have an impact on the adoption of project management approaches of Western origin. Most software development companies in Mauritius have not been able to adopt such approaches. Some researchers have even raised their concerns on the need to have an indigenous solution for managing projects in the developing world.

Furthermore, software projects have certain inherent strengths and weaknesses that make them unique as compared to traditional engineering projects. Most methodologies/models/standards are generic in nature and do not take advantage of these aspects. Soft skills form part of aspects that have been neglected in the management of projects. A software project is even more sensitive to the need for such skills due to the lack of skilled human resources in the developing countries. A project manager or software development staff who is not satisfied with the environment provided in an organisation may leave for a better organisational climate.

This research has also shown that much improvement is required to increase the level of project success. The project maturity level of software development organisations in Mauritius and South Africa, for instance, show that more effort need to be made to develop the sector.

As a result of all the problems and different settings in developing countries, it is expected that an indigenous project management maturity model like $ESPM^3$ will improve the level of software project success.

13.7 Future Work

As mentioned above, $ESPM^3$ was applied successfully to two projects (discussed in chapter 12). Given that all the KPAs were applied so as to manage the projects, it could be noted that $ESPM^3$ was not applied in an optimised manner for the two organisations although the projects were successful. The evolution of maturity level 2 KPAs has to be pursued in the first instance until they effectively attain maturity. After an organisation has been assessed to be at maturity level 2, the evolution of maturity level 3 KPAs has to be considered until it attains maturity. Assessment is again carried out to determine whether the organisation has reached the required maturity level. The continuous process improvement group of KPAs will then ensure that continuous improvement is undertaken by the organisation. Therefore, an organisation adopting ESPM^3 will have to pursue a progressive development of the KPAs by ensuring that it has reached maturity level 2 before undertaking steps to attain maturity level 3.

The progressive development of an organisation's project management approach should ensure that organisations can benefit from achieving organisational project management maturity (Project Management Institute, 2004b). As a correlation has been established between project management maturity and project success (Schiltz, 2003; Sonnekus and Labuschagne, 2004), it is worthwhile for organisations in Mauritius to adopt a project management model like ESPM^3. Not only does the model take on board the specific requirements associated with the development of a software product but also takes into account the situation prevailing in the country with respect to social, political, economic, cultural and related conditions.

ESPM^3 will have to be applied to some more developing countries to show its success. In future, the model will have to be tested in software development organisations in other developing countries to justify its suitability and provide confidence in its use. Organisations that are not using any methodology/model/standard need to be encouraged to adopt ESPM^3.

As ESPM^3 provides a simple means of assessment of the project management maturity level of software development oganisations, another avenue to be explored would be to gauge the maturity level of the industry in various developing countries. The maturity level of the software industry will, therefore, provide an indication on the improvement required to develop the IT sector in various developing countries.

A software tool based on ESPM^3 that may assist in the implementation of software projects may add to the benefits of using the model. The software tool will need to incorporate applicable KPAs and help the organisation to pursue with their evolution towards the different maturity levels. The standard inputs and outputs for each KPA will be available and the project managers may add more depending on the project. Improved control of the KPAs may be achieved by tracking their PDCA cycles in an integrated manner. The software tool has to issue alerts regarding various deviations that occur during project execution.

Links with various tools and other software tools may be provided. The model may be customised to incorporate the required tools for cost estimation. For instance, the use of object point analysis may be incorporated as a standard cost estimation tool. Similarly Work Breakdown Structures may also be included so as to help in reducing the complexity of a proposed software product. A reports generating feature may also be taken care of in order to provide the project manager with information on the progress of projects.

It is, therefore, believed that a software tool based on $ESPM^3$ that would automate the implementation of the model could assist organisations in further improving software project delivery. The project manager will be able to save useful time with such a tool. It is sometimes mentioned that methodologies/models/standards for project management encourage so much bureaucracy and rigidity that organisations may get bogged down in red tape (Herbsleb et al., 1997). Therefore, the software tool may reduce these challenges and allow the project manager with more time to address other problems that may occur during the execution of software projects.

The weakness noted in the second case study regarding user resistance may probably be solved using Joint Application Development (JAD). JAD allows a team comprising the software developers and the users for whom the system is to be built. In this way, both the users and the software developers may take an active role to help in the success of the project. Therefore, the use of JAD with the $ESPM^3$ may be investigated in the future.

In addition, the use of prototypes in projects may be considered so that ambiguities and communication problems may be resolved at the initial stage of development. Different prototypes (for example evolutionary prototypes and throwaway prototypes) may be used and an evaluation could be carried out to determine the effects.

However, it is foreseen that the adoption of $ESPM^3$ for developing countries may improve the successful delivery of software projects. The development of a suitable software tool that can automate the use of the model may assist project managers further. In addition, future work can be undertaken by testing the software tool in other developing countries. Feedbacks from other developing countries can help in improving the model.

References

ACCENTURE. 2006. *Accenture in Africa: Our Changing Landscape*. Website. http://www.accenture.com/NR/rdonlyres/DEAEC-FCC-54DE-457F-A31D-615C5A83839E/0/africaReport_final.pdf. 25th September 2008.

ALADWANI, A. M. 2002. IT Project uncertainty, planning and success: An empirical investigation from Kuwait. *Information Technology and People*. 15 (3), 210-226.

ARORA, S. 2003. *Soft skills training: Outsourced vs in-house*. Website. http://www.expressitpeople.com/20040426/cover.shtml. 3rd February 2005.

BABER, L.R. 1987. The Spine of Software – Designing Provably Correct Software: Theory and Practice. John Wiley & Sons Ltd. Chichester.

BACH, J. 1994. The Immaturity of CMM. Website. http://www.satisfice.com/articles/cmm.htm. 30th March, 2005.

BELZER, K. 2004. *Project Management: Still More Art than Science*. Website. http://www.pmforum.org/library/papers/BusinessSuccess.htm. 19th August 2004.

BOODHUN, V. 2005. *A Systematic Approach to Software Project Management in a Government IT Department*. Master of Science project. University of Mauritius.

BPO SECRETARIAT. 2005. Survey on the ITES BPO sector in Mauritius. Board of investment

BROOKS, F.P. 1995. The mythical Man-month: Essays on software engineering. Addison-Wesley: Reading, MA, US.

BRUCE, A. and LANGDON, K. 2000. *Project Management*. Dorling Kindersley: New York.

CARAYANNIS, E.G.; KWAK, Y.H. and ANBARI, F.T. 2003. The Story of Managing Projects: An Interdisciplinary Approach. Quorum Books.

CCTA. 1997. PRINCE 2 Pocketbook. Central Computer and Telecommunications Agency.

CCTA. 2002. *Central Computer and Telecommunications Agency. Project Management industry initiatives*. Website. http://www.ogc.gov.uk/prince/progmgtbrief1997.pdf. 12th December 2002.

CHAPMAN, A. 2001. *Stress management*. Website. http://www.businessballs.com/stressmanagement.htm. 24th November 2004.

CHATURVEDI, S. 2005. *Strategic Human Resource Management*. Website. http://www.humanlinks.com/manres/articles/shrm.htm. 23rd August 2005.

CLELAND, D.I. 1999. Project Management: Strategic Design and Implementation. McGraw-Hill. Singapore.

CLELAND, D.I. 2004. The Evolution of Project Management. IEEE Transactions on Engineering Management. 51(4). 396-397.

CMCS. 2009. PMO Solution. Website. http://www.cmcs.ae/en/services/pmo-solution.html. 28th February 2009.

CMMI PRODUCT TEAM. 2002. *Capability Maturity Model Integration*. Website. http://cmsu2.cmsu.edu/public/classes/sam/cis4655/cmmi.pdf. 3rd January 2006.

COOKE-DAVIES, T. 2002. *Project Management Maturity Models. Does it make sense to adopt one?* Project Management Today. Website. http://www.humansystems.net/papers/TCDarticles/MAYTCDweb.pdf. 8th October 2008.

DOBBINS, R. and PETTMAN, B.O. 1997. Self-development: the nine basic skills for business success. *Journal of management development*. 16(8), 521.

EID, A. 2002. Cyber Island, Silicon Valley under the Tropics. *Islander*. 32. 66-67.

ENTERPRISE PLANNING ASSOCIATES. 1998. *Project management maturity Level: Interactive Quick Look*. Website. http://www.sqi.gu.edu.au/sc7/mirror/N1401-N1450/07n1406.pdf. 1st April 2006.

EUROMETHOD. 2002a. *Background and history of the Euromethod project*. Website. http://projekte.fast.de/Euromethod/history.htm. 31st May 2002.

EUROMETHOD. 2002b. *Euromethod*. Website. http://projekte.fast.de/Euromethod. 27th June 2002.

FARAGO, J. and SKYRME, D. 1995. *The Learning Organization*. Website. http://www.skyrme.com/insights/3lrnorg.htm. 24th November 2004.

FITZERALD, J. and DENNIS, A. 2002. Business Data Communications and Networking. John Wiley & Sons. USA.

GARDINER, P.D. and STEWART, K. 2000. Revisiting the golden triangle of cost, time and quality: the role of NPV in project control, success and failure. *International Journal of Project Management*. 18(4), 251-256.

GIDO, J. and CLEMENT, J.P. 1999. Successful Project Management. SouthWestern Publishing Co.

GOVERNMENT PORTAL. 2006. *Empowering the community to make Mauritius a regional IT hub*. Website. http://www.gov.mu/portal/site/Mainhomepage/?content_id=2c667005d8b0a010VgnVCM1000000a04a8c0RCRD. 20th September 2006.

GRAY, C.F. and LARSON, E.W. 2000. *Project Management: The Managerial Process*. McGraw-Hill: Singapore.

HARRISON, M.; SWEENEY, M; TAYLOR, L. and WOOD, A. (2003). *Measuring project management maturity: a comparison sampling of project managers in Huntsville, Ala: Best Practices*. Website. http://www.findarticles.com/p/articles/mi_m0KAA/is_5_32/ai_111506022. 10th April, 2006.

HARVEY, D. and BROWN, D.F. 2006. An experiential approach to organisation development. Pearson Prentice Hall: Upper Saddle River, New Jersey.

HAYNES, M.E. 1989. *Project Management: From idea to implementation*. Kogan Page. London.

HEFNER, R. 2003. *Transition from SW-CMM to CMMI:Yes or No?* Website. http://www.uces.csulb.edu.SPIN/media/pdf/03SoCASPIN_CMMI_Transition_052203b.ppt. 18th January, 2006.

HERBSLEB, J.; ZUBROW, D.; GOLDENSON, D.; HAYES, W. and PAULK, M. 1997. Software Quality and the Capability Maturity Model. *Communications of the ACM*. 40(6). 30-40.

HILLS, C.W.L. 2005. *International Business: Competing in the global marketplace*. Tata McGraw-Hill.

HOFSTEDE, G. 1983. Cultural dimensions for project management. *International Journal of Project Management*. 1(1), 41-48.

HORNBY, A.S. 1995. Oxford Advanced Learner's Dictionary. Oxford University Press.

HUBER, N. 2003. *Hitting targets? The state of UK IT project management*. Website. http://www.computerweekly.com/Articles/2003/11/03/198355/more-it-projects-are-hitting-their-targets.htm. 5th September 2007.

HUGHES, B. and COTTERELL, M. 2006. *Software Project Management*. McGraw-Hill: London, UK.

INTERNATIONAL INSTITUTE FOR LEARNING. 2002. *Kerzner's project management maturity level*. Website. http://www.iil.com/project_management_training/project_management_maturity_model.asp. 10 April 2006.

JALOTE, P. 2002. *Software Project Management in Practice*. Addison-Wesley.

JAMES, D. 1999. *Managing People in Organisations*. Financial World Publishing.

KEY SKILLS LTD. 2003. *What is PRINCE 2.* Website. http://www.prince2.com/whatis p2.html. 2nd December 2003.

KIRSCH, L.J. 2004. *Software Project Management: An integrated perspective for an emerging paradigm*. Website. http://www.pinnaflex.com/pdf/framing/CH15.pdf. 19th August 2004.

KORTON, D. 1980. Community organisation and rural development: a learning process approach. *Public Administration Review*. September/October 1980.

KOZAK-HOLLAND, M. 2007. *History of Project Management*. Website. http://www.lessons-from-history.com/Level 2/History_of_PM_page.html. 29th June 2007.

KWAK, Y.H. 2003. *Brief History of Project Management*. Website. http://home.gwu.edu/~kwak/PM_History.pdf. 19th September 2005.

KWAK, Y.H. and IBBS, C.W. 2002. Project Management Process Maturity $(PM)^2$ Model. *Journal of Management in Engineering*. 150-155.

LUSSIER, R.N. 1990. *Human Relations in Organisations*. Homewood, IL: Irwin.

MACKY, A. 2008. *Mauritius: Resolving the food crisis*. Website. http://allafrica.com/stories/200807230728.html. 14th October 2008.

MANAGEMENT AUDIT BUREAU. 2005. *ISO 9001 Implementation*. Website. http://www.gov.mu/portal/site/mabsite/menuitem.b4fa590d611ae5b397da5810 65c521ca/. 4th January 2006.

MAURITIAN STANDARD. 2001. Quality management systems – Requirements. Mauritius Standards Bureau.

MCDONALD, J. 2001. Why is Software Project Management Difficult? And What that Implies for Teaching Software Project Management. Computer Science Education.11(1). 55-71.

MICROSOFT. 2007. *Microsoft Project for Windows Version History*. Website. http://support.microsoft.com/kb/98026. 27th June 2008.

MINISTRY OF IT & TELECOMMUNICATIONS. 2008. Website. http://www.gov.mu/portal/site/telecomit. 19th September 2008.

MOORE, J. 2002. *ISO 12207 and Related Software Life-Cycle Standards*. Website. http://www.acm.org/tsc/lifecycle.html. 3rd September 2002.

MULLALY, M. E. 2001. *IT Projects: Built to Last*. Website. http://www.gantthead.com/article.cfm?ID=16329. 18th February 2005.

MUNRO-FAURE, L.; MUNRO-FAURE, M. and BONES, E. 1994. *Achieving Quality Standards*. Pitman Publishing.

MURIITHI, N. and CRAWFORD, L. 2003. Approaches to project management in Africa: implications for international development projects. *International Journal of Project Management*. 21(5), 309-319.

NCB. 2008. *National Computer Board*. Website. http://www.gov.mu/portal/sites/ncbnew/main.jsp. 18th September 2008.

NEL, P.S.; GERBER, P.D; VAN DYK, P.S.; HAASBROEK, G.D.; SCHULTZ, H.B.; SONO, T. and WERNER, A. 2001. *Human Resources Management*. South Africa: Oxford University Press.

NEWSTROM, J. W. and DAVIS, K. 2001. Organisational Behaviour: Human Behaviour at Work. New York: McGraw-Hill.

NICHOLAS, J.M. 2001. *Project Management for Business and Technology*. Prentice-Hall of India.

NIENABER, R. and CLOETE, E. 2003. A Software Agent Framework for the Support of Software Project Management. Proceedings of the 2003 annual research conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology. ACM Digital Library.

NOE, R.A.; HOLLENBECK, J.R.; GERHART, B. and WRIGHT, P.M. 2004. *Human Resource Management*. New York: McGraw-Hill.

OFFICE OF GOVERNMENT COMMERCE. 2004. Prince 2 Maturity Model. Draft V0.05.01.doc.

OHASHI HIGH TECHNOLOGY CORP. 2007. Website. http://www.ohashi-ht.com/Japanese%20Business.html. 20th March 2007.

O'LEARY, T.J. and O'LEARY, L.I. 2000. Computing Essentials. McGraw-Hill: New York, US

PARTHASARATHY, M.A. 2008. *Practical Software Estimation: Size, Effort, and Scheduling of Projects*. Website. http://www.informit.com/articles/article.aspx?p=705249&seqNum=2. 18th September 2008.

PAULK, M.C; CURTIS, B.; CHRISSIS, M.B. and WEBER, C.V. 1993. *Capability maturity model for software*. Software Engineering Institute. Carnegie Mellon University.

PHILLIPS, J. 2002. *IT Project Management*. Tata McGraw-Hill: New Delhi, India

PMBOK GUIDE®. 2006. *A Guide to the Project Management Body of Knowledge*. Third Edition. Website. http://www.pmi.org/info/PP_PMBOKGuideThirdExcerpts.pdf. 9th March 2006.

PM SOLUTIONS. 2004. *What is Project Management Maturity*. Website.

http://www.pmsolution.com/maturitymodel/whatispmm.htm. 27th July 2004.

POLITICS OF MAURITIUS. 2006. Website. http://en.wikipedia.org/wiki/Politics_of_Mauritius. 4th March 2006.

PRIMAVERA SYSTEMS. 1995. *Primavera Project Planner (P3)*. Website. http://www.p2pays.org/ref/01/00047/2-03.htm. 25th May 2008.

PROEHL, R. A. 2001. *Organisational change in Human Services*. UK: Sage Publication.

PROJECT MANAGEMENT EXPERTS. 2004. *What is Project Management Maturity*. Website. http://www.pmsolution.com/maturitymodel/whatispmm.htm. 27th July 2004.

PROJECT MANAGEMENT INSTITUTE. 2004a. *Organisational Project Management Maturity Model: OPM3 Overview*. PMI, Inc. Pennsylvania, USA.

PROJECT MANAGEMENT INSTITUTE. 2004b. *An Executive Guide to OPM3: Transform Strategy into Results*. Website. http://opm3online.pmi.org. 5th April 2006.

PUBLIC PROCUREMENT OFFICE. 2008. Website. http://www.gov.mu/portal/site/pposite/index.htm. 22nd September 2008.

RAO, M.T. 2004. Key issues for global IT sourcing: Country and individual factors. *Information Systems Management*, 21(3), 16-21.

REHMAN, A. and HUSSAIN, R. 2007. Software Project Management Methodologies/Frameworks Dynamics “A Comparative Approach”. Information and Emerging Technologies, 2007 ICIET 2007. 1-5.

RONAN, N.J. 1993. Developing the African Manager: The good, the bad and the competent. Management education and development. 24(4). 388-394.

RONDINELLI, D. A. 1976. Why development projects fail: Problems of project management in developing countries. *Project management quarterly*. 7(7), 10-15.

SANTROCK, J.W. 2000. *Psychology*. Congress Lib.

SAUER, C. and CUTHBERTSON, C. 2003. *The State of IT Project Management in the UK 2002-2003*. Templeton College, University of Oxford

SCHACH, S.R. 2002. *Object-Oriented and Classical Software Engineering*. McGraw-Hill. New York.

SCHWALBE, K. 2004. *Information technology project management*. Course Technology, Thomson Learning: Cambridge, MA, US.

SCHILTZ, S.J. 2003. *A practical method for assessing the financial benefit of project management*. Master of Science thesis. City University. Website. http://www.pmi-switzerland.ch/knowledge/cost ofbadpm_schiltz_v11.pdf. 19th May 2004.

SOFTWARE ENGINEERING INSTITUTE. 2005. *List of Published SCAMPI Appraisal Results*. Website. http://seir.sei.cmu.edu/pars/pars_list_iframe.asp. 30th March 2006.

SONNEKUS, R. and LABUSCHAGNE, L. 2004. Establishing the relationship between IT project management maturity and IT project success in a South African context. *Proceedings of the 2004 PMSA Global Knowledge Conference.*

SPICE PROJECT. 1995. *ISO/IEC Software Process Assessment: A model for process management*. Website http://www.sqi.gu.edu.au/sc7/mirror/N1401-N1450/07n1406.pdf. 1st April 2006.

STANDISH GROUP. 2004. *Third Quarter Research Report*. Website. http://www.standishgroup.com/sample_research/PDFpages/q3-spotlight.pdf. 9th March 2006.

SIL. 2006. *State Informatics Limited*. Website. http://www.stateinformatics.com. 18th September 2008.

STUCKENBRUCK, L.C. and ZOMORRODIAN, A. 1987. Project management: the promise for developing countries. *International Journal of Project Management*. 5(3), 167-175.

SUKHOO, A.; BARNARD, A.; ELOFF, M.M. and VAN DER POLL, J.A. 2004a. Towards a framework for evaluating strengths and weaknesses of software projects. *Proceedings of the 2004 PMSA Global Knowledge Conference*. 193-199.

SUKHOO, A.; BARNARD, A.; ELOFF, M.M. and VAN DER POLL, J.A. 2004b. A survey of project management tools, techniques and methodologies used in Mauritius: The current status. *Proceedings of the 2004 PMSA Global Knowledge Conference*. 320-329.

SUKHOO, A.; BARNARD, A.; ELOFF, M.M. and VAN DER POLL, J.A. 2005a. An assessment of Software project management maturity in Mauritius. *Informing Science Institute journal*.

SUKHOO, A.; BARNARD, A.; ELOFF, M.M.; VAN DER POLL, J.A. and MOTAH, M. 2005b. Accommodating soft skills in software project management. *Informing Science Institute journal*.

SUKHOO, A.; BARNARD, A.; ELOFF, M.M. and VAN DER POLL, J.A. 2007. An evolutionary software project management maturity model for Mauritius. *Interdisciplinary Journal of Information, Knowledge, and Management*. 2. 99-118.

SUPIĆ, H. 2005. Project Management Maturity of Selected Organizations in Croatia. 8th International Conference on Telecommunications – Contel 2005.

SWANEPOEL, B.J., SLABBERT, J.A., ERASMUS, B.J. and BRINK, M. 1999. The Management of Employment Relations. Johannesburg: Butterworths.

TATNALL, A. and SHACKETON, P. 1995. Project Management Software – the Fourth Tool? Software Education Conference. IEEE. 320-325.

TURNER, J. R. 1993. *The handbook of project-based management: improving the processes of achieving strategic objectives*. London:McGraw-Hill.

University of Portsmouth. 2002. *Project Management Methodologies*. Website. http://www.sis.port.ac.uk/~allangw/pm-mthds.htm. 31th May 2002.

US DEPARTMENT OF STATE. 2007. *Mauritius: 2007 Investment Climate Statement – Mauritius*. Website. http://www.state.gov/e/eeb/ifd/2007/80726.htm. 25th September 2008.

VAN DER POLL, H.M. and VAN DER POLL, J.A. 2007. Towards an Analysis of Poor Learner Performance in a Theoretical Computer Literacy Course. Proceedings of the 2007 Informing Science and IT Education Joint Conference. 85-95.

WEHMEIER, S. 2005. Oxford Advanced Learner's Dictionary.

WHITE, D. and FORTUNE, J. 2002. Current Practice in Project Management – an empirical study. *International Journal of Project Management*. 20(2002), 1-11.

WIDEMAN, R. M. 2002. Comparing Prince2® with PMBOK®. Vancouver: AEW Services.

WIDEMAN, R. M. 2005. PMBOK Guide, Third Edition: *Is more really better?* Website. http://maxwideman.com/papers/pmbok3/pmbok3b.pdf. 3rd January 2006.

12MANAGE. 2006. *Prince 2*. Website. http://www.12manage.com/methods_ccta_prince2.html. 5th April 2006.

APPENDICES

APPENDIX A

Summary of Sequential Project Management Methodologies/Models/Standards

Summary of sequential project management methodologies/models/standards

Project Management Body of Knowledge PMBOK®

The first publication of PMBOK® by the Project Management Institute (PMI) dates back to 1983. The 1996 release with over 150 pages provided a realistic and mature guide to project management. The worldwide audience resulted from the availability of the 1996 release being distributed freely from the PMI's website. Over the years, the number of members increased considerably and new versions of PMBOK® were released.

PMBOK® is organised into nine knowledge areas, grouped into core functions (scope, time, cost and quality) and facilitating functions (human resource, communication, risk and procurement) with integration management tying them together (Sonnekus and Labuschagne, 2004) as shown diagrammatically in figure A.1.

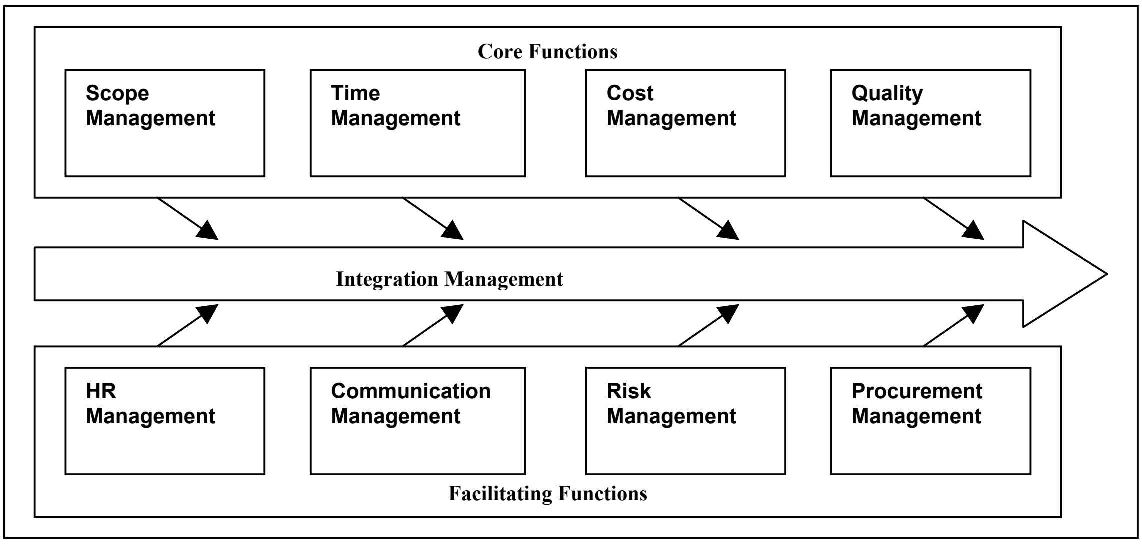


Figure A.1: PMBOK® knowledge areas (Adapted from Schwalbe, 2004)

Each knowledge area is divided into a number of processes as shown below:

Core Functions		
	Scope Management	Initiation
		Scope planning
		Scope definition
		Scope verification
		Scope change control
	Time Management	Activity definition
		Activity sequencing
		Activity duration estimating
		Schedule development
		Schedule control
	Cost Management	Resource planning
		Cost estimating
		Cost budgeting
		Cost control
	Quality Management	Quality planning
		Quality assurance
		Quality control
Integration Management		Plan development
		Plan execution
		Overall change control
Facilitating Functions		
	HR Management	Organisation planning
		Staff acquisition
		Team development
	Communications management	Communications planning
		Information distribution
		Performance reporting
		Administrative closure
	Risk Management	Risk identification
		Risk quantification
		Risk response development
		Risk response control
	Procurement management	Procurement planning
		Solicitation planning
		Solicitation
		Source selection
		Contract administration
		Contract close-out

Within each process, various inputs are transformed into outputs using some tools and techniques. The processes are in turn mapped onto five process groups, which are namely initiating, planning, executing, controlling and closing.

Euromethod

Euromethod (EM) has been in use since November 1989 as version 0 within the European Union (EU) (Euromethod, 2002a). Its development has been the result of efforts for mutual understanding between customers and suppliers of information systems. EM version 1 was launched in July 1996. In order to promote EM outside the EU, it is now marketed under "Information Services Procurement Library" – ISPL (Euromethod, 2002b).

Methodology

According to Hughes and Cotterell (2006), an organisation can be perceived as having a number of information systems (IS) and each one may need to be changed from time to time from one state (initial state) to another (final state). Each change is implemented by an adaptation, which transforms an information system from a defined initial state including knowledge into a defined final stage (figure A.2).

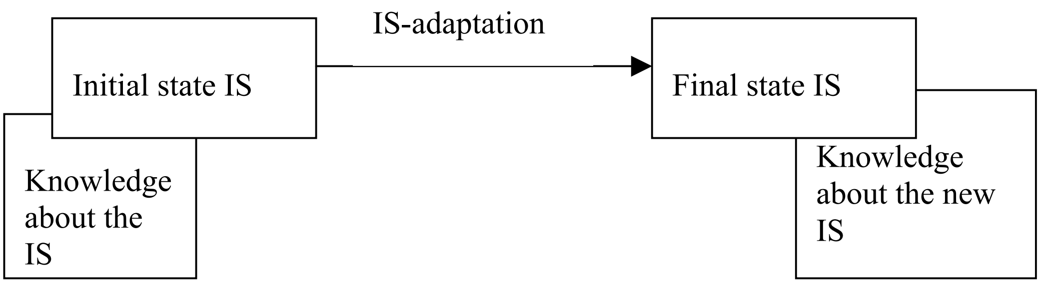


Figure A.2: Information System adaptation

However, in a large organisation, a sequence of adaptations may need to be applied and the final state of one adaptation may become the initial state of the next adaptation. Each adaptation may be subject to a procedure process, whereby a contract will be agreed upon between a supplier and customer.

The principles of Euromethod

EM can be used in any information system involving:

(i) a customer,
(ii) a supplier (external or internal to the organisation), and
(iii) a contract (formal or informal) between the customer and the supplier.

EM is based on a number of basic principles, namely:

(i) Information System adaptation.

Changes in ways of conducting businesses compel information systems to adapt. The procurement and development of IS's are addressed with a broad consideration for products and processes.

At product level, organisation, human resources and technical elements are considered. These resources are required in processes. EM, with a wider scope than software development, aims at improving an information system.

At the process level, EM addresses changes in IS through modification and automation in order to meet the needs of the organisation.

(ii) Variety of contracts.

EM is suitable for IS's having an initial and final state. This applies to contractual arrangements during the life cycle of an information system.

The contractual arrangements are shown in figure A.3.

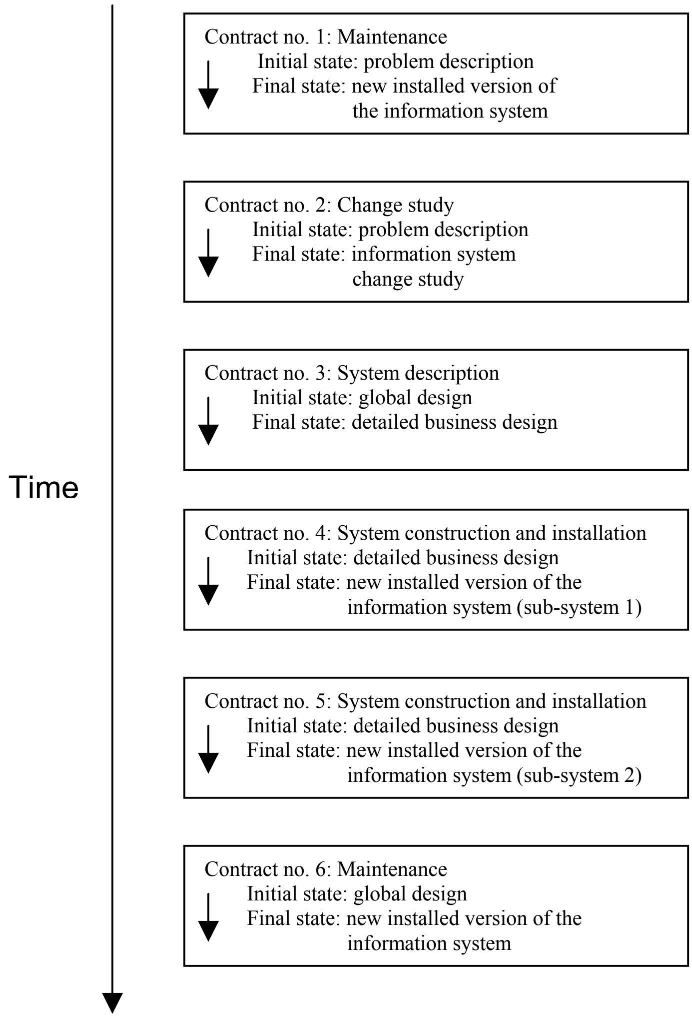


Figure A.3: Contractual arrangements

During the life of an information system, several IS-adaptations regulated by a contract will occur, example one or more systems description, construction, installation, or maintenance projects. When a contract is established, the information system and the knowledge about the information system are in a state called the initial state of the adaptation. At the termination of the contract, the new state of the information system and its accompanied knowledge becomes the final state.

(iii) Customer-Supplier relationships.

EM focuses mainly on the contractual level, starting from the tendering, through the signing of a contract and proceeding with the production of a set of deliverables until the contract is terminated. Throughout these processes, EM aims at:

(a) enhancing the customer-supplier relationships.
(b) bridging the cultural difference between the parties concerned (supplier and customer).

(iv) Situation driven.

EM applies a situation-driven approach to Information System (IS) adaptation. An IS-adaptation strategy is designed for the reduction or containment of risks inherent in the problem situation and it places specific requirements on the delivery plan of the contract. The situation-driven approach is shown in figure A.4.

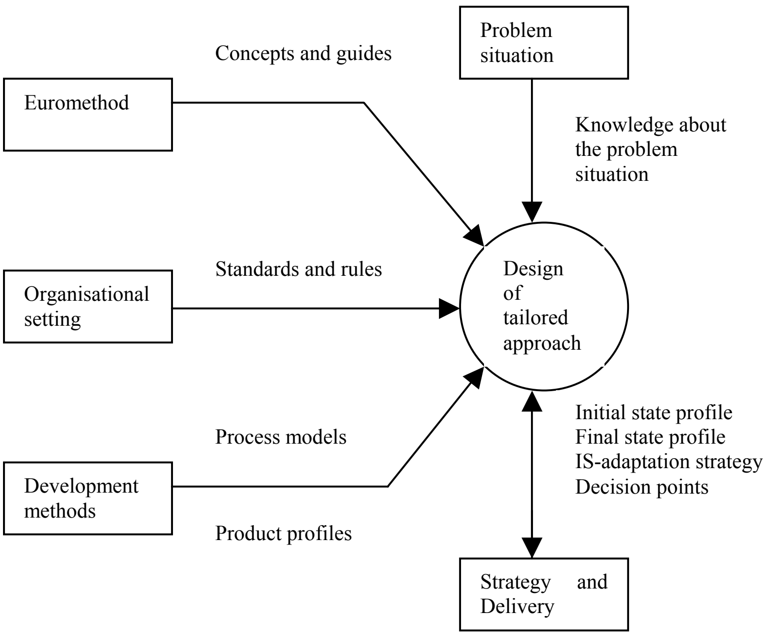


Figure A.4: Situation-driven approach

(v) Focus on decisions.

Three processes are identified during execution of transactions between a customer and a supplier, namely:

(a) a tendering process, which starts with a call for tender and ends with the signing of the contract.

(b) a production process, where deliverables are produced.

(c) a completion process, where the contract is technically and commercially terminated.

During these processes, decisions are taken by the customer in order to determine the properties of the new information system and the approval about the way it is produced.

(vi) Focus on deliverables.

EM focuses more on deliverables rather than on the supplier's activities involved in producing the deliverables (figure A.5).

Classification of the deliverables is based on a distinction between target domain and project domain.

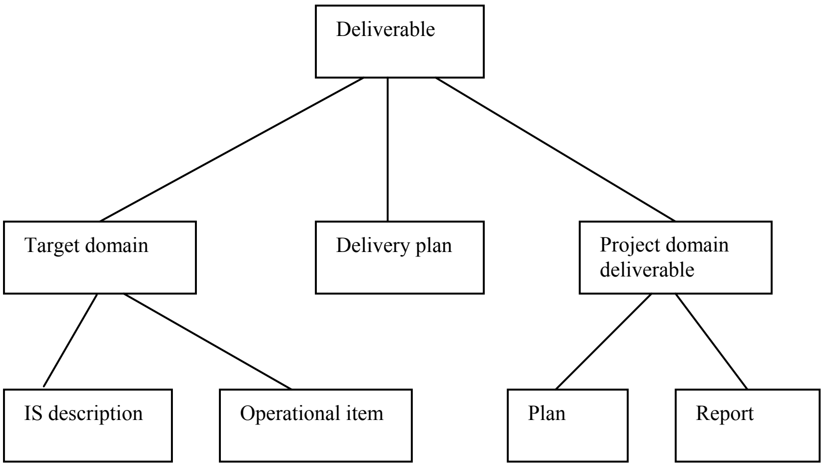


Figure A.5: Deliverables

(vii) Method bridging

EM alone does not guide the design of a tailored approach. It, however, integrates well in the organisational setting, which will have its own standards and rules to be followed and development methods to be applied.

EM, therefore, offers to bridge its own concepts as well as the concepts of the development methods.

BS 6079: 1996

BS6079 ‘Guide to project management’ was published as a 49-page document in 1996 (Hughes and Cotterell, 2006). It is a set of guidelines rather than a standard. It has an engineering rather than IT background.

BS 6079 elements

The main elements of BS 6079 are:

(i) the project cycle (figure A.6).

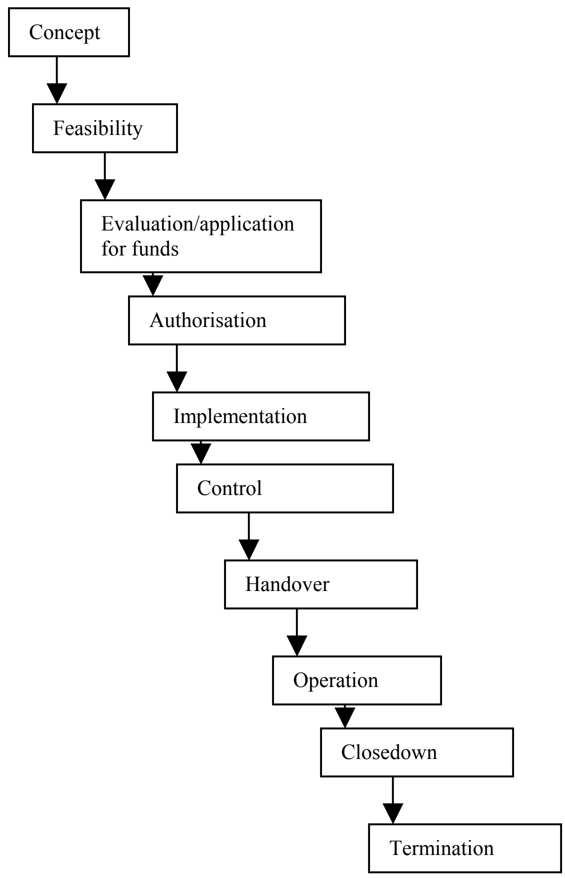


Figure A.6: Project cycle

BS 6079 views a project as a system life cycle. The actual development of the project starts after the feasibility study has established that it is worthwhile and ends when the system is operational.

(ii) project organisation.

A project is considered as a set of activities that cut across the normal functional structures of most organisations. A matrix management structure is desirable. Explanation is provided on how to bring about an organisational change although this may take 3 to 5 years.

BS 6079 emphasises on task owners, whereby each task has an owner, who is held accountable for its successful completion.

(iii) planning process.

According to BS 6079, a plan should contain five main elements, namely:

(a) introduction
(b) commitment acceptance
(c) work breakdown structure (WBS)
(d) schedule
(e) statement of work (SOW)

The activity-based approach of BS 6079 emphasises on a WBS right from the beginning of the project. In addition, a statement of work is created for each element in the WBS.

(iv) project control.

This exercise is carried out through:

(a) change control.

Although no detail steps are provided, an outline of the requirements is followed.

(b) financial control.

Disbursement of funds may be made as and when progress is achieved. However, funding must be allowed for contingencies.

(v) monitoring progress

Reports from task owners are analysed at this stage. Risk assessment should be continuously made and use of techniques based on statistical probability is recommended.

(vi) Supporting Techniques.

Calculation of the net present value and the analysis of the earned value are encouraged at this stage.

ISO 12207

ISO 12207 has been developed specifically for software development. It focuses on the characteristics of good documentation as the lack of documentation is always an issue complained of by software developers, but its availability may result in reluctance to be read.

ISO 12207 perceives a software development project as an Information System (IS). A project comprises of a set of activities. Each activity passes information to the next activity. The effectiveness of the IS determines the success or failure of the project.

Methodology (Hughes and Cotterell, 2006)

Five processes are identified and shown in figure A.7.

(i) Acquisition.
(ii) Supply.

(iii) Development.

(iv) Operation.

(v) Maintenance.

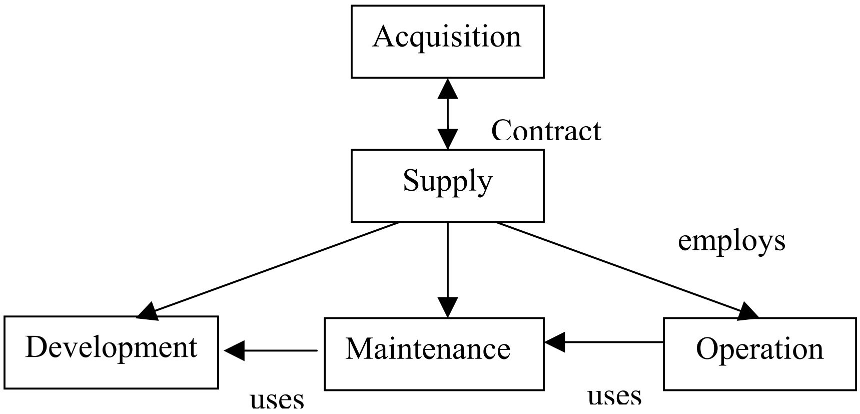


Figure A.7: Acquisition process

Clearly, maintenance and operation processes belong to the post-implementation phase. Therefore, acquisition, supply and development processes will be considered.

Acquisition Process

This is the set of procedures that the customer has to go through in order to obtain the requested software from a supplier. The acquisition process consists of five main activities:

(i) Initiation activity, where the requirements of the customer are described, defined and the best way to acquire a software is considered.

(ii) Request for proposal (RFP).

A “request for proposal” document is prepared during this activity. The topics in the document are as follows:

- System requirements.
- Scope statement.
- Instructions for bidders.
- List of software products.
- Control of sub-contracts.
- Technical constraints.

(iii) Contract preparation and update.

This activity involves the determination of criteria to be used in the selection of a supplier. Once the supplier is selected, the final contract can be negotiated between the supplier and the customer.

(iv) Supplier monitoring.

This activity is performed through joint reviews, audit, verification and validation.

(v) Acceptance and completion of contract.

When the final product is delivered, acceptance tests are performed by the customer before the software enters the operational stage.

Supply Process

The following activities are concerned:

(i) Initiation.

This activity begins with the supplier's decision to bid after receiving the RFP, that is decision to bid is taken.

(ii) Preparation of response.

The response is prepared after consultation of people with relevant experience.

(iii) Contract negotiation.

A successful proposal will lead to the negotiation and signing of a contract between the supplier and the customer.

(iv) Planning.

A detailed plan of the work to be carried out, based on the requirements of the RFP, is submitted. The contents of the plan are as follows:

(a) project organisational structure.
(b) development environment including facilities, standards, procedures and tools.
(c) work breakdown structure.
(d) management of quality characteristics.
(e) management of safety, security and other critical requirements.
(f) sub-contractor management.
(g) verification and validation.
(h) acquirer involvement.
(i) user involvement.
(j) risk management.
(k) security policy.
(l) arrangements for obtaining any regulatory approvals.
(m) scheduling, tracking and reporting.
(n) Training.

(v) Execution and control.

This activity involves the execution of the plan. The supplier must constantly control the process of the work and at the same time ensure that the product quality does not suffer. Compliance of subcontractors with requirements must also be by the supplier during this activity.

(vi) Review and evaluation.

Review of progress and evaluation of work by the customer must be authorised as per contract.

(vii) Delivery and completion.

Delivery plan and post-delivery support have to be carefully considered as per plans initially agreed upon.

The Development Process

Several activities are concerned here as shown in figure A.8.

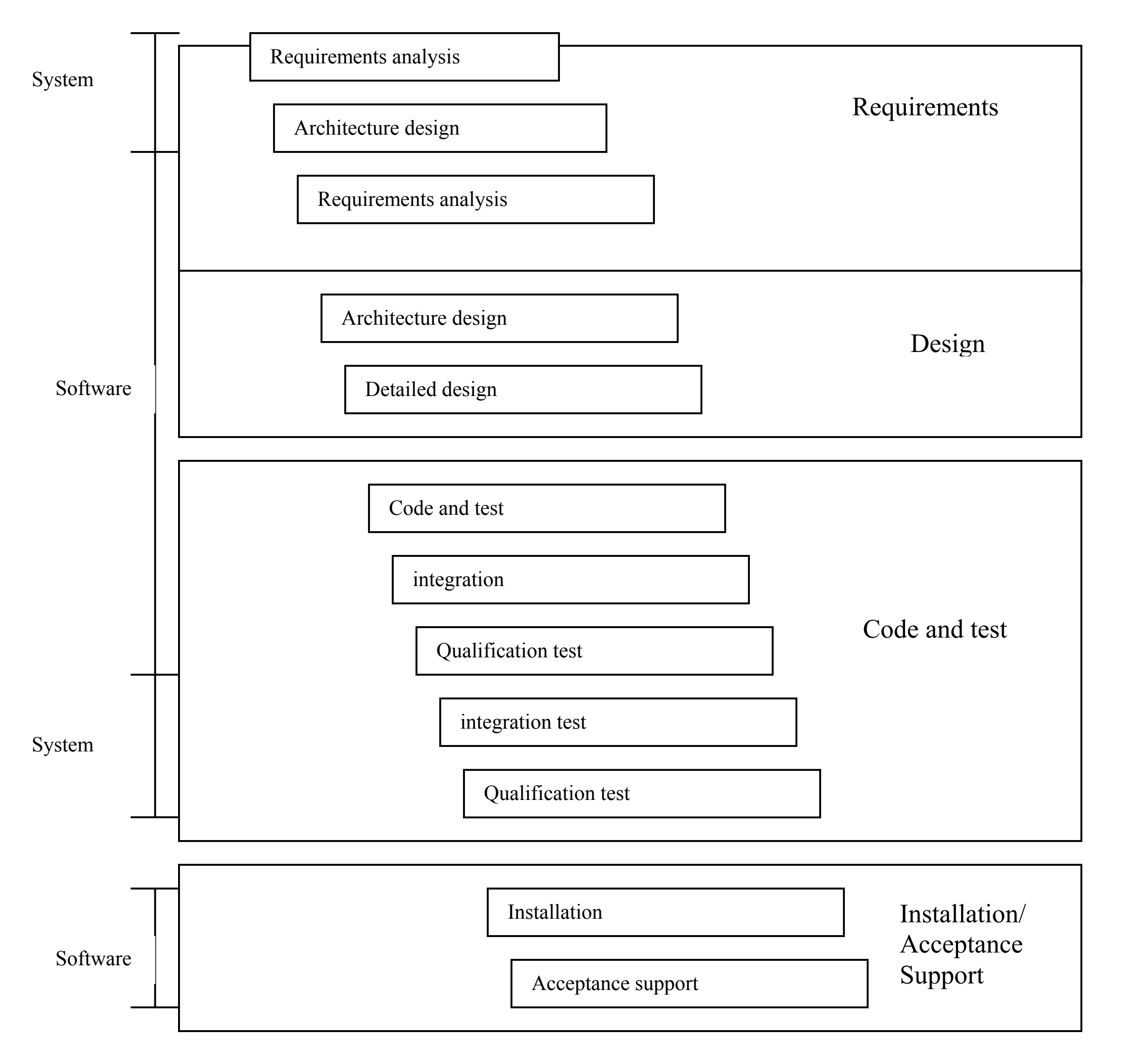


Figure A.8: Development process

PRINCE 2®

PRINCE 2® (PRojects IN Controlled Environments) is a structured project management methodology for managing projects in a logical, organised way, following defined steps. PRINCE 2® was launched in 1996 and is now the de-facto standard for project management being used in both private and public sectors in the United Kingdom for all types and size of projects (CCTA, 1997).

Methodology

The methodology as per PRINCE 2® Documentation/manuals, covers the following processes:

- (i) Directing a Project
- (ii) Starting up a Project
- (iii) Initiating a Project
- (iv) Managing Stage Boundaries
- (v) Controlling a Stage
- (vi) Managing Product Delivery
- (vii) Closing a Project
- (viii) Planning

The processes are interrelated as shown in figure A.9.

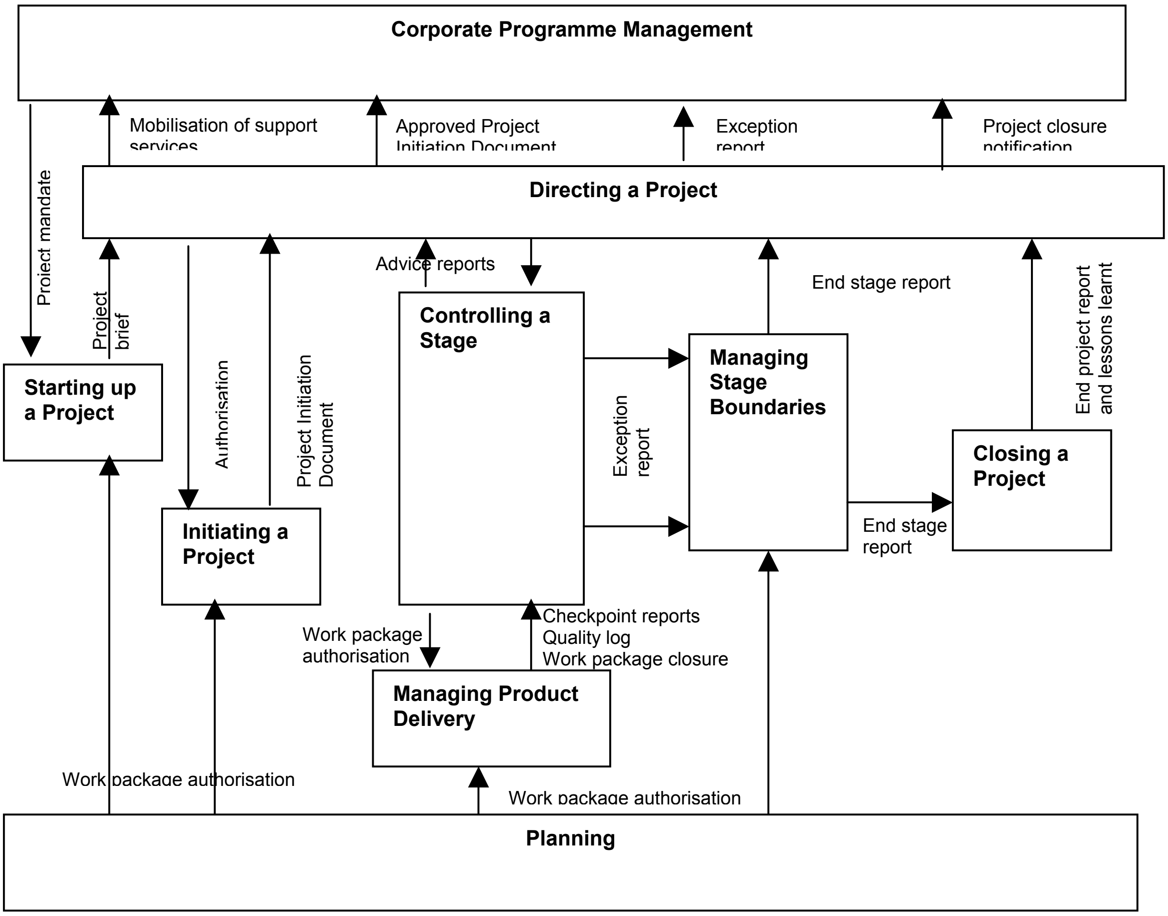


Figure A.9: PRINCE 2® processes (CCTA, 1997)

Directing a Project (DP)

Directing a Project runs from the start-up of the project until its closure. This process is aimed at the Project Board. The Project Board manages by exception, monitors via reports and controls through a number of decision points.

The key processes for the Project Board break into four main areas:

- Initiation (starting the project).
- Stage boundaries (commitment of more resources after checking results so far).
- Ad hoc direction (monitoring progress, providing advice and guidance, reacting to exception situations).
- Project closure (confirming the project outcome and controlled close).

This process does not cover the day-to-day activities of the project manager.

Starting up a Project (SU)

This is the first process in PRINCE 2®. It is a pre-project process, designed to ensure that the pre-requisites for initiating the project are in place. The process expects the existence of a Project Mandate, which defines in high-level terms the reason for the project and what outcome is sought. Starting up a Project should be very short.

The work of the process is built around the production of three elements:

- Ensuring that the information required for the project team is available.
- Designing and appointing the Project Management Team.
- Creating the Initiation Stage Plan.

Initiating a Project (IP)

The objectives of Initiating a Project are to:

- Agree whether or not there is sufficient justification to proceed with the project.
- Establish a stable management basis on which to proceed.
- Document and confirm that an acceptable Business Case exists for the project.
- Ensure a firm and accepted foundation to the project prior to commencement of the work.
- Agree to the commitment of resources for the first stage of the project.
- Enable and encourage the Project Board to take ownership of the project.
- Provide the baseline for the decision-making processes required during the project's life.
- Ensure that the investment of time and effort required by the project is made wisely, taking account of the risks to the project.

Managing Stage Boundaries (SB)

This process provides the Project Board with key decision points on whether to continue with the project or not. The objectives of the process are to:

- Assure the Project Board that all deliverables planned in the current Stage Plan have been completed as defined.
- Provide the information needed for the Project Board to assess the continuing viability of the project.
- Provide the Project Board with information needed to approve the current stage's completion and authorise the start of the next stage, together with its delegated tolerance level.
- Record any measurements or lessons that can help later stages of this project and /or other projects.

Controlling a Stage (CS)

This process describes the monitoring and control activities of the Project Manager involved in ensuring that a stage stays on course and reacts to unexpected events. The process forms the core of the Project Manager's effort on the project, being the process that handles the day-to-day management of the project. Throughout a stage there will be a cycle consisting of:

- Authorising work to be done.
- Gathering progress information about that work.
- Watching for changes.
- Reviewing the situation.
- Reporting.
- Taking any necessary corrective action.

This process covers these activities, together with the on-going work of risk management and change control.

Managing Product Delivery (MP)

The objective of this process is to ensure that planned products are created and delivered by:

- Making certain that work on products allocated to the team is effectively authorised and agreed upon.
- Ensuring that work conforms to the requirements of interfaces identified in the Work Package.
- Ensuring that the work is done.
- Assessing work progress and forecasts regularly.
- Ensuring that completed products meet quality criteria.
- Obtaining approval for the completed products.

Closing a Project (CP)

The purpose of this process is to execute a controlled close to the project. The process covers the Project Manager's work to wrap up the project either at its end or at premature close. Most of the work is to prepare input to the Project Board to obtain its confirmation that the project may close.

The objectives of closing a project are, therefore, to:

- Check the extent to which the objectives or aims set out in the Project Initiation Document have been met.
- Confirm the extent of the fulfilment of the Project Initiation Document and the Customer's satisfaction with the deliverables.
- Obtain formal acceptance of the deliverables.
- Ensure to what extent all expected products have been handed over and accepted by the Customer.
- Confirm that maintenance and operation arrangements are in place (where appropriate).
- Make any recommendations for follow-on actions.
- Capture lessons resulting from the project and complete the Lessons Learned Report.
- Prepare an End Project Report.
- Notify the host organisation of the intention to disband the project organisation and resources.

Planning (PL)

Planning is a repeatable process, used by the other processes whenever a plan is required. The process makes use of the PRINCE 2® product-based planning technique and covers:

- Designing the plan.
- Defining and analysing the plan's products.

- Identifying the necessary activities and dependencies.
- Estimating the effort required.
- Scheduling resources.
- Analysing the risks.

PRINCE 2® Components

The following components can be identified:

(i) Plans

The Project Plan summarises the resource requirements and costs. The Stage Plan contains details for day-to-day control by the Project Manager.
The Team Plan is prepared when there are a number of teams working in a stage.

(ii) Controls

PRINCE 2® provides a series of controls throughout the project life cycle in order to ensure that deviations are within accepted limit.

(iii) Stages

A project is divided into stages, which are manageable. In order to proceed to the next stage, approval of the Project Board is required.

(iv) Risk

The Project Manager is responsible to ensure that risks are identified, recorded and regularly reviewed.

(v) Quality

The project delivered has to be of an acceptable level of quality.

(vi) Configuration Management

The purpose of configuration management is to identify, track and protect the project's products.

(vii) Change

A change control mechanism captures all issues and track them to completion.

(viii) Organisation

This is achieved through roles. A role can be allocated to one person, shared between a number of people or combined with another role.

Principles

PRINCE 2® focuses on product deliverables in contrast with other methodologies, which aim at carrying out activities. Throughout its life cycle, PRINCE 2® emphasises on the business case of a project.

A business case is used to define the information that justifies the setting up, continuation or termination of the project. It answers the question 'Why should the project be done?'(PRINCE 2® Documentation/Manuals).

APPENDIX B

Questionnaire for supplier and user survey in Mauritius

ASSESSMENT OF PROJECT MANAGEMENT METHODOLOGY

Dear Sir/Madam,

I am presently following a PhD course at the University of South Africa (UNISA). During the course of my study, I am carrying out a survey, which is focusing on the need for the development of a software project management methodology for developing countries. The results will help in framing a new methodology oriented at better responding to the information needs of software development companies in developing countries.

Your answers are very important to the accuracy of this research. All answers will be confidential and will be used only in combination with those of other respondents. Therefore, information that may be published will be solely summarised statistics.

Kindly return the completed questionnaire to:

Mr. A. Sukhoo
Email Address: aneeravsukhoo@yahoo.com

Thanking you for your collaboration.

Please fill in the name of your country below

____

PARTICULARS OF YOUR ORGANISATION

1. What is the name of your organisation [*optional*]?

   ____

2. Which title best describes your current position? (please tick the most appropriate choice)

   - ❑ Director/General Manager
   - ❑ IT Manager
   - ❑ Systems Analyst
   - ❑ Programmer
   - ❑ Other (please specify below)

   ____

3. What is the size of your organisation? (please indicate the approximate number of staff below)

   IT Professionals ____________ Non-IT Professionals __________________

4. What is the annual turnover for IT projects in your organisation? (please indicate the approximate turnover below)

   ____

5. What is the main activity of your organisation? (please tick all that apply)

   - ❑ Project management
   - ❑ Software development
   - ❑ Hardware sales
   - ❑ IT Consultancy
   - ❑ Other (please specify below)

   ____

PROJECT MANAGEMENT PARTICULARS

6. Which of the following Project Management Methodologies have you heard of (please tick all that apply)?:

   - ❑ PRINCE 2
   - ❑ Euromethod

- ❑ ISO 12207
- ❑ PMBOK
- ❑ BS6079

7. Do you use any Project Management Methodology (please tick the most appropriate choice)

- ❑ Yes ____________________(please specify the name)
- ❑ No Skip to 10

8. Which of the following is/are addressed by the Project Management Methodology in use in your organisation(please tick all that apply):

- ❑ Planning
- ❑ Cost Management
- ❑ Quality Management
- ❑ Human Resource Management
- ❑ Communications Management
- ❑ Risk Management
- ❑ Procurement Management

9. Does the Project Management Methodology in your organisation support multi-project management?

- ❑ Yes
- ❑ No

10. What is the average number of concurrent projects handled by a project manager? (please indicate the average number below)

____

11. What Project Management software tools (e.g. Primavera or one developed within your organisation) are used in your organisation?

____

12. Which of the following Project Management tools are used in your organisation (please tick all that apply):

- ❑ Work Breakdown Structure (WBS)
- ❑ Gantt Chart
- ❑ Project Evaluation and Review Technique/Critical Path Method (PERT/CPM)
- ❑ Gantt with links
- ❑ Risk Assessment Matrix
- ❑ Monte Carlo simulation
- ❑ Earned Value
- ❑ Responsibility Matrix
- ❑ Resource Estimate

- ❑ Progress Report
- ❑ Other (please specify below)

____

13. Which of the following approach is being used in your organisation for software development (please tick all that apply):

    - ❑ Classical System Development Life Cycle (e.g. based on the Waterfall model)
    - ❑ Structured System Development Life Cycle
    - ❑ Fourth Generation Language (4GL) techniques and approach (e.g. use of screen designer, report writer, application generator, data dictionary and data manipulation, non-procedural database query language, etc.)
    - ❑ Prototyping
    - ❑ Other (please specify below)

    ____

14. What is the quality management standard used in your organisation? (please tick the most appropriate choice)

    - ❑ International Standards Organisation (ISO)
    - ❑ Total Quality Management (TQM)
    - ❑ a standard developed in-house
    - ❑ no standard is used

15. What is the approximate percentage of projects (annually) in your organisation that suffers from failure to meet deadline (please tick the most appropriate choice)

    - ❑ above 75%
    - ❑ between 50% and 75% (both inclusive)
    - ❑ above 0% but less than 50%
    - ❑ 0%

16. What is the approximate percentage of projects (annually) in your organisation that suffers from budget overrun (please tick the most appropriate choice)

    - ❑ above 75%
    - ❑ between 50% and 75% (both inclusive)
    - ❑ above 0% but less than 50%
    - ❑ 0%

17. What is the approximate percentage of projects (annually) in your organisation that suffers from failure to satisfy user requirements annually (please tick the most appropriate choice)

- ❑ above 75%
- ❑ between 50% and 75% (both inclusive)
- ❑ above 0% but less than 50%
- ❑ 0%

18. Indicate your level of agreement/disagreement on whether the following factors contribute to deviation in project achievement. Please add other reasons which were relevant to your organisation and tick where appropriate.

LEGEND

SA	: Strongly Agree	D	: Disagree
A	: Agree	SD	: Strongly Disagree
N	: Neutral		

	Reasons for deviation in project achievement	**SA**	**A**	**N**	**D**	**SD**
1.	No formal methodology is used					
2.	Bad planning					
3.	User resistance					
4.	Inadequate or improper communication infrastructure (between project team and customer)					
5.	Inappropriate project team					
6.	No Executive support					
7.	Failure of project team to understand user needs					
8.	Inadequate or absence of user involvement					
9.	Risk management					
10.	Difficulty to cope with concurrent projects					
11.	Lack of expertise in innovative/new technology					
12.	Lack of skilled professionals					
13.	Mobility of labour					

19. Do you agree that Project Management Methodologies developed by Western countries are applicable for developing countries?

- ❑ Yes
- ❑ No
- ❑ Partially agree

20. If your answer above (question 19) is "No" or "Partially agree", list some factors that justify the inappropriateness (or partial inappropriateness) of these methodologies.

____

____

____

21. Do you believe that some form of indigenous project management should prevail in developing countries?

- ❑ Yes
- ❑ No

Please fill in your name and email address below for any further queries and to receive a copy of the results. However, the information is *optional*.

Name ____

Email Address ____

END OF QUESTIONNAIRE

ASSESSMENT OF SOFTWARE

Dear Sir/Madam,

I am presently following a PhD course at the University of South Africa (UNISA). During the course of my study, I am carrying out a survey, which is focusing on the need for the development of a software project management methodology for developing countries. The results will help in framing a new methodology oriented at better responding to the information needs of software development companies in developing countries.

Your answers are very important to the accuracy of this research. All answers will be confidential and will be used only in combination with those of other respondents. Therefore, information that may be published will be solely summarised statistics.

Kindly return the completed questionnaire to:

Mr. A. Sukhoo
Email Address: aneeravsukhoo@yahoo.com

Thanking you for your collaboration.

1. How do you rate the cost of software purchased by your organisation? (please tick the most appropriate choice)

- ☐ very expensive
- ☐ expensive
- ☐ just right
- ☐ cheap

2. How would you rate user-friendliness of software developed for your organisation? (please tick the most appropriate choice)

- ☐ very good
- ☐ good
- ☐ satisfactory
- ☐ poor

3. How would you rate the performance of software developed for your organisation? (please tick the most appropriate choice)

- ☐ very good
- ☐ good
- ☐ satisfactory
- ☐ poor

4. How would you rate the security features of software developed for your organisation? (please tick the most appropriate choice)

- ☐ very good
- ☐ good
- ☐ satisfactory
- ☐ poor

5. Are you satisfied with the time taken by developers to deliver software? (please tick the most appropriate choice)

- ☐ yes
- ☐ no

6. What, in your opinion, is/are the cause(s) of failure of software to meet expected objectives? (please tick all that apply)

- ☐ lack of communication between suppliers and customers
- ☐ lack of experience on the part of software developers
- ☐ unprofessional attitude of suppliers
- ☐ possible lack of best practices for software development and IT management
- ☐ All the above are not applicable

7. Will you be more inclined to purchase software from a vendor who has an internationally recognised Quality Management System implemented? (please tick the most appropriate choice)

☐ Yes

☐ No

8. Do your software suppliers have an internationally recognised Quality Management System implemented? (please tick the most appropriate choice)

☐ Yes

☐ No

☐ I am not aware

√ Please indicate your level of agreement or disagreement with the following statements by placing a tick in the most appropriate box .

The following may enhance customer satisfaction with respect to software:

LEGEND

SA	: Strongly Agree	D	: Disagree
A	: Agree	SD	: Strongly Disagree
N	: Neutral		

		SA	A	N	D	SD
9.	improving communication between suppliers and users through regular meetings or informal discussion.					
10.	frequent participation of users in the software development process by providing regular feedback.					
11.	suppliers should consider seriously user satisfaction rather than just abiding strictly by the specifications agreed upon initially on award of a contract.					
12.	a chief information officer (CIO) in charge of issues pertaining to IT should be appointed on the user side.					
13.	each user department must recruit its own software developers, IT manager, systems analysts, Database Administrators, etc.					
14.	more effort should be devoted to elimination of errors in software.					

END OF QUESTIONNAIRE
THANKING YOU FOR YOUR COOPERATION

APPENDIX C

Summary of project management maturity models

Project Management Maturity Models (PMMMs)

1. Introduction

Project management is supported by project management approaches, some of which are based on maturity models. Organisations achieve progressive development of its project management approach, using both qualitative and quantitative data to make decisions. Just as individuals benefit from achieving personal maturity, organisations can benefit from achieving organisational project management maturity (Project Management Institute, 2004b).

Various PMMMs have been developed (e.g. MicroFrame self-assessment tools; Project Management Process Maturity $(PM)^2$; Kerzner's maturity model; and SEI's Capability Maturity Model Integration; Prince 2 Maturity Model; OPM3^{TM}). Most of them are defined by five maturity levels, namely 1 through 5.

Project Management Maturity Models can be categorised as generic models and software specific models as shown in figure C.1.

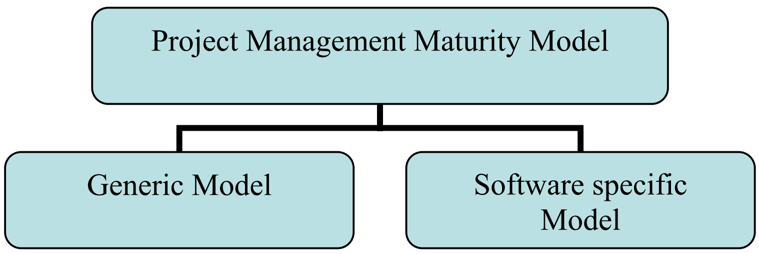


Figure C.1: Project management maturity models

Generic models are used for all types of projects while the software specific models are used for management of software projects. Software specific models take care of specificities inherent in software projects. The following sections summarise some of the models that exist.

2. Berkeley Project Management Process Maturity Model $(PM)^2$

Kwak and Ibbs (2002) proposed a 5-level project management maturity $(PM)^2$ model to assess and improve an organisation maturity level (figure C.2). Each level breaks project management processes and practices into nine knowledge areas (Integration, Scope, Time, Cost, Quality, Human Resource, Communications, Risk and Procurement) and five phases (Initiate, Plan, Execute, Control and Close out), thus adopting the classification of the Project Management Body of Knowledge (Kwak and Ibbs, 2000). This is shown in figure C.3.

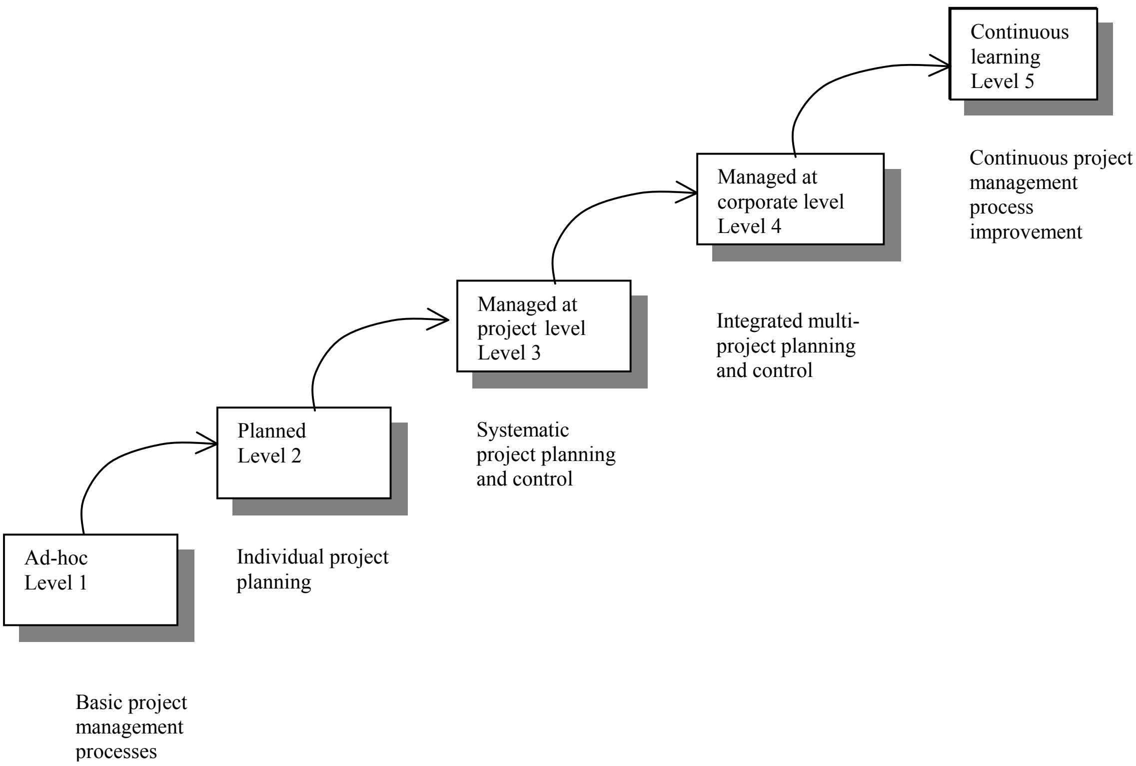


Figure C.2: Project management process maturity model (Kwak and Ibbs, 2002)

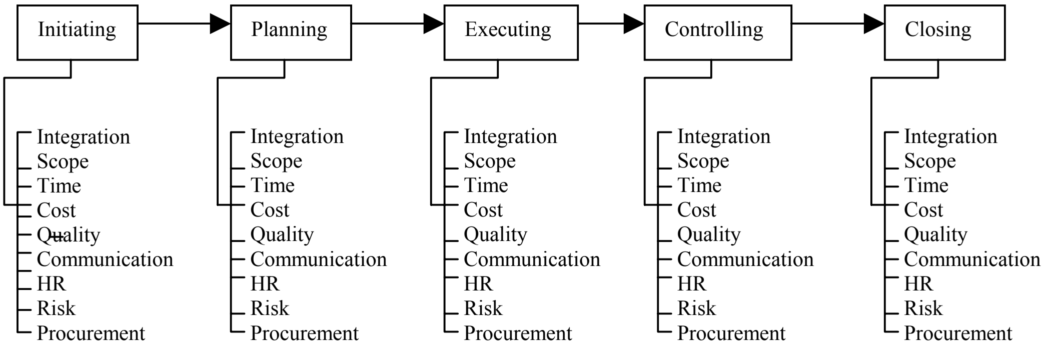


Figure C.3 Integrating project processes and project management knowledge areas (Kwak and Ibbs, 2002)

Level 1 – Ad-hoc stage

No project management processes or practices are consistently available.

Level 2 – Planned stage

Informal processes are defined and informal problems are identified at this level.

Level 3 – Managed stage

Formal project planning and control systems are managed.

Level 4 – Integrated stage

Multiple projects are managed efficiently. Processes data are quantitatively analysed, measured and stored.

Level 5 – Sustained stage

Project management processes are fully understood and continuously improved.

Shortcomings:

(i) Political, socio-economic and cultural aspects are not considered.
(ii) Soft skills are not considered.
(iii) The model is a generic one and therefore it does not address specific problems inherent in software project management.

2. Microframe's Self Assessment Tool

The Project Management Maturity Model was developed by Micro Frame Technologies, Inc. and Project Management Technologies, Inc. (Enterprise Planning Associates, 1998). A self-assessment tool comprising 50 multiple choice questions was designed.

According to Enterprise Planning Associates (1998), the model is a process and system framework for helping organisations to improve their project management and management of projects functions. It provides a phased set of maturity descriptions, improvement criteria, operating metrics, and questions that can be used to assess the current level of maturity and develop a focused plan for improving the effectiveness of project and functional management. The objective of the model is to improve the ability of organisations to consistently achieve goals for cost, schedule, technical, and customer satisfaction.

The model emphasizes incremental improvement in single and multi-project management disciplines and ranks maturity across five levels, as follows:

Level 1 - Ad-Hoc

The project management process is described as disorganised, and occasionally even chaotic. Systems and processes are not defined. Project success depends on individual effort. Projects are prone to chronic cost and schedule problems.

Level 2 - Abbreviated

Some project management processes and systems are established to track cost, schedule, and performance. Underlying disciplines, however, are not well understood or consistently followed. Project success is largely unpredictable and cost and schedule problems are the norm.

Level 3 - Organized

Project management processes and systems are documented, standardized, and integrated into an end-to-end process for the company. Project success is more predictable. Cost and schedule performance is improved.

Level 4 - Managed

Detailed measures of the effectiveness of project management are collected and used by management. The process is understood and controlled. Project success is more uniform. Cost and schedule performance conforms to plan.

Level 5 - Adaptive

Continuous improvement of the project management process is enabled by feedback from the process and from piloting innovative ideas and technologies. Project success is the norm. Cost and schedule performance is continuously improving.

Shortcomings:

(i) Political, socio-economic and cultural aspects are not considered.
(ii) Soft skills are not considered.
(iii) The model is a generic one and therefore it does not address specific problems inherent in software project management.

3. PRINCE 2 Maturity Model

According to the Office of Government Commerce (2004), PRINCE 2 Maturity Model has the purpose of enabling organisations to gauge, by assessment, their maturity in the use of the model. Each maturity level has “Key Process Areas” and “Key Practices”. The latter describes what an organisation has to do in order to manage projects effectively. The PRINCE 2 Maturity Model assesses maturity of organisations at three levels as follows:

Level 1 - Initial

At this level, the organisation recognises projects and runs them differently to its ongoing business. No key process areas exist at the initial level.

Level 2 - Repeatable

At this level the model is used across the organisation, but inconsistently. Eleven key process areas can be identified at the repeatable level and they are namely:

- (i) Directing a project.
- (ii) Initiating a project.
- (iii) Controlling a stage.
- (iv) Closing a project.
- (v) Business case.
- (vi) Organisation.
- (vii) Plans/planning.
- (viii) Management of Risk.
- (ix) Controls.
- (x) Quality management.
- (xi) Configuration management and change control.

Level 3 – Defined

PRINCE 2® is applied consistently across the organisation. The following key process areas are defined at this level:

(i) Organisational focus.
(ii) Tailoring of PRINCE 2®.
(iii) Training in PRINCE 2®.
(iv) Integrated management.
(v) Quality assurance.

Shortcomings:

(i) Political, socio-economic and cultural aspects are not considered.
(ii) Soft skills are not considered.
(iii) Given that PRINCE 2 maturity model evolved from PRINCE 2®, people management and contract management are not covered (12Manage, 2006).
(iv) The model is a generic one and therefore it does not address specific problems inherent in software project management.

4. Organizational Project Management Maturity Model (OPM3™)

The Project Management Institute (2004a) defines organisational project management as the application of knowledge, skills, tools and techniques to organisational and project activities to achieve the aims of an organisation through projects or the systematic management of projects, programs and portfolios in alignment with the achievement of strategic goals. OPM3™ is a maturity model developed by the Project Management Institute for organisations to assess their maturity as well as to increase their maturity when applicable. The model is aligned with the PMBOK®.

OPM3™ has three interlocking elements, namely Knowledge, Assessment and Improvement. The Knowledge element provides the body of best practices for users. The Assessment element allows comparison of the organisation's project management practice against the best practices in OPM3™. The Improvement element allows organisations to make use of the assessment results to move ahead with change initiatives. These elements are applied in a cycle called the OPM3™ cycle as shown below:

Shortcomings:

(i) Political, socio-economic and cultural aspects do not take into consideration the specific needs of developing countries.
(ii) Soft skills are not considered.
(iii) The model is a generic one and therefore it does not address specific problems inherent in software project management.

5. Kerzner's Management Maturity Model

Dr. Harold Kerzner developed a generic project management maturity model comprising 5 levels. Project management is regarded as a core competency that many companies must develop in order to remain competitive in the market (Schiltz, 2003). The various levels are depicted in figure C.4.

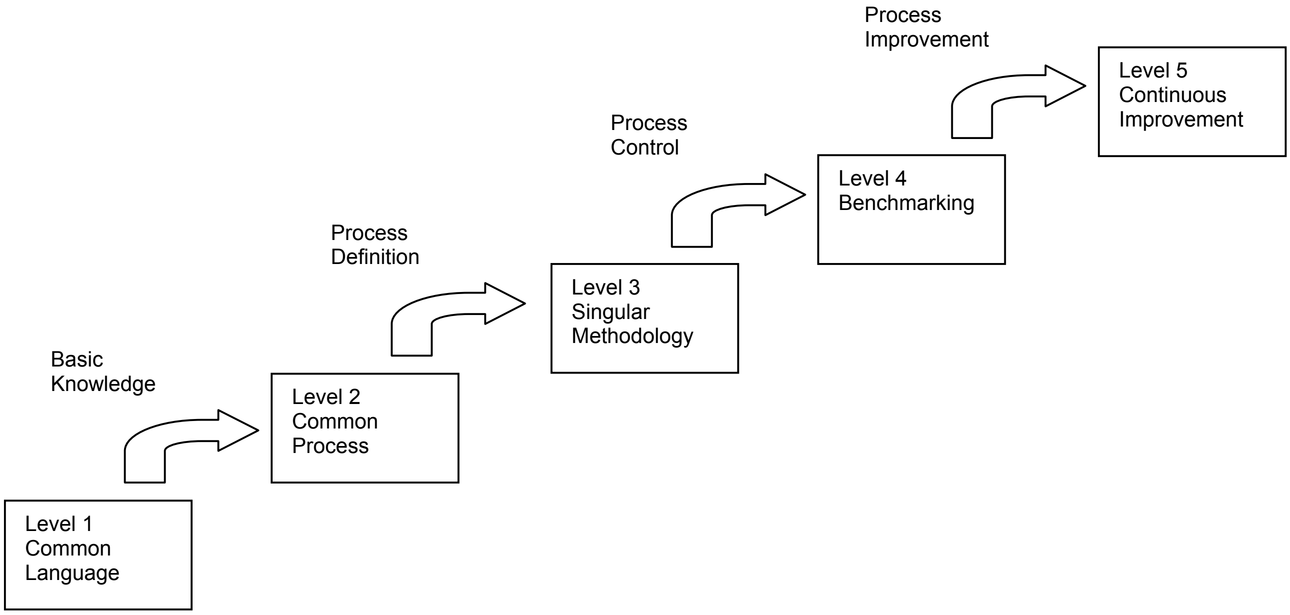


Figure C.4: Kerzner's project maturity levels (adapted from Harrison, Sweeney, Taylor and Wood, 2003)

According to Harrison, Sweeney, Taylor and Wood (2003), the different levels are described as follows:

Level 1 – Common Language

Common Language is the basic knowledge of PM and the terminology used. Eight management categories are defined at this level and they are namely:

- Scope/Integration.
- Time.
- Cost.
- Human Resource.
- Procurement.
- Quality.
- Risk.
- Communication.

Level 2 – Common Processes

Common Processes defined and developed are applicable and repeatable. Assessment for Level 2 is divided into five life cycle phases:

- Embryonic-basic recognition of the need for and benefits of PM.
- Executive Management Acceptance-executive support in utilising PM.
- Line Management Acceptance-line/functional management support in utilising PM.
- Growth-development of a PM methodology.
- Initial Maturity development and usage of a cost and schedule control system; ongoing PM education/training program.

Level 3 – Singular Methodology

Singular Methodology is the synergistic effect of combining all corporate methodologies. At this level, disjointed processes and methodologies are merged into a single methodology, which can be measured by the following six traits:

- Integrated Processes-multiple possible processes have been combined into one
- Cultural Support-corporate culture encourages collaborative efforts to promote Project Management
- Management Support-each level of management supports the methodology in their roles
- Informal Project Management-methodology supported by informal means
- Training and Education-acknowledgement that PM training and education reaps rewards
- Behavioural Excellence-training for PM to replace line management mentality.

Level 4 – Benchmarking

Benchmarking process improvement is required to maintain a competitive advantage. Benchmarking can be a powerful tool for assessing cycle time, quality, resource allocation, training practices, sales productivity, and other business-related issues. In order for a benchmarking program to succeed, it must evaluate the right metrics, measure those metrics accurately and relevantly, and report the metrics clearly in a timely fashion.

Level 5 – Continuous Improvement

Continuous Improvement evaluates the enhancement to Project Management from each improvement. An organisation must meet four key Level 4 requirements in order to advance to Level 5:

- Create an organisation dedicated to benchmarking.
- Develop a project management benchmarking process.
- Decide what to benchmark and against whom to benchmark.
- Recognise the benefits of benchmarking.

Shortcomings:

(i) Political, socio-economic and cultural aspects do not take into consideration the specific needs of developing countries.
(ii) Soft skills are not considered.
(iii) It is a generic methodology and does not address specifically software project management.

6. SEI's Capability Maturity Model Integration (CMMI)

The Capability Maturity Model Integration (CMMI) was developed by the Software Engineering Institute (SEI). It evolved from the (Capability Maturity

Model) CMM. Instead of using multiple CMMs, the CMMI Product Team combined three models, namely Capability Maturity Model for Software (SW-CMM), Electronic Industries Alliance Interim Standard (EIA/IS) and Integrated Product Development Capability Maturity Model (IPD-CMM) into a single framework for enterprise-wide process improvement. The CMMI provides an easy migration from SW-CMM to CMMI.

The key components of the CMMI are process areas and these are grouped into maturity levels. There are 25 process areas in total. Each process area has specific and generic goals. Specific goals have specific practices. Generic goals have common features that organise generic practices (figure C.5).

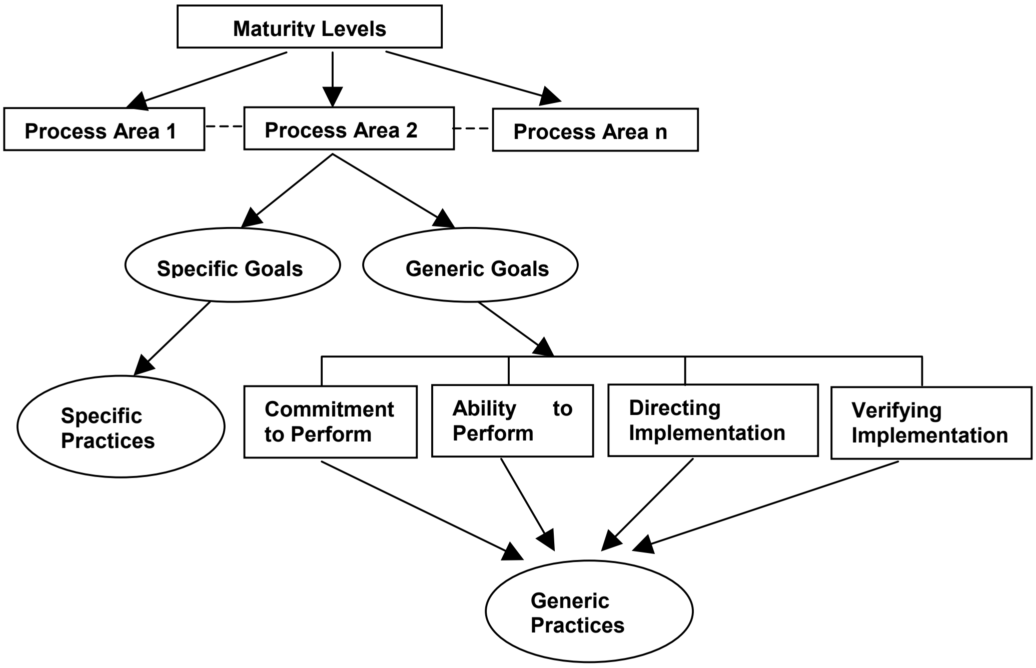


Figure C.5: CMMI Model Components (CMMI Product Team, 2002)

CMMI has five maturity levels as shown in figure C.6. Maturity level 2 onwards has process areas which are focus areas in order to achieve a particular maturity level.

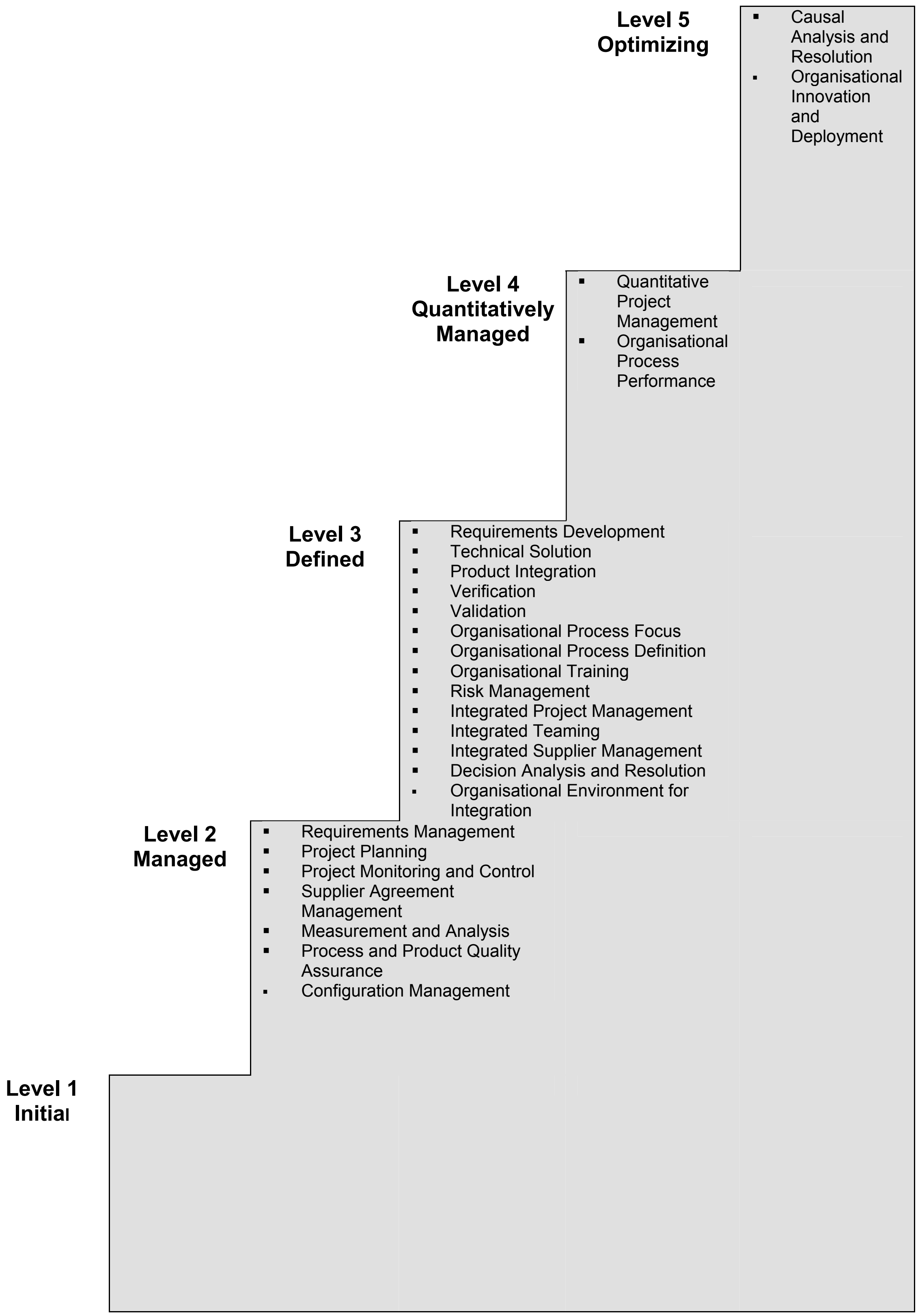


Figure C.6: CMMI Model (Hefner, 2003)

Level 1 – Initial

At this level, processes are usually ad-hoc and chaotic. The success of projects depends on the skills of the project members. Although organisations at level 1 often produce products that work, they frequently exceed budget and schedule. During a crisis, organisations tend to abandon processes and are not able to repeat past successes.

Level 2 – Managed

At this level, processes are planned, performed, measured and controlled. All the specific and generic goals are achieved for the process areas concerned at the level. Plans are documented.

Level 3 – Defined

Processes are well characterised and understood, and are described in standards, procedures, tools and methods. All processes are consistent across the organisation except where tailoring of guidelines are required.

Level 4 – Quantitatively Managed

Quantitative objectives for quality and process performance are established. Processes are qualitatively and quantitatively predictable.

Level 5 – Optimizing

This level focuses on continually improving process performance through incremental and innovative technological improvements.

Shortcomings of CMMI:

(i) the adoption of CMM encourages so much bureaucracy and rigidity that organisations may get bogged down in red tape (Herbsleb, Zubrow, Goldenson, Hayes and Paulk, 1997). CMMI does not seem to have improved on this issue.

(ii) CMM requires the experience of very knowledgeable people to succeed (Bach, 1994). Training of staff and the help of consultants are required to implement CMMI. One organisation implementing CMMI in Mauritius had to invest a lot on training and consultancy. This investment is not possible for small or even medium sized organisations

(iii) Political, socio-economic and cultural aspects are not considered in CMMI

(iv) The documentation on CMMI is voluminous as the guide contains 639 pages. It may take much time for an organisation to be conversant with the framework without the help of consultants

(v) 139 organisations adopting CMMI are at level 5 and 30 organisations are certified at level 4. The distribution of 583 organisations using CMMI in the world is shown figure C.7. The number of organisations adopting CMMI is still low in the world and a greater percentage is below level 4.

(vi) Soft skills are not treated as a process area.

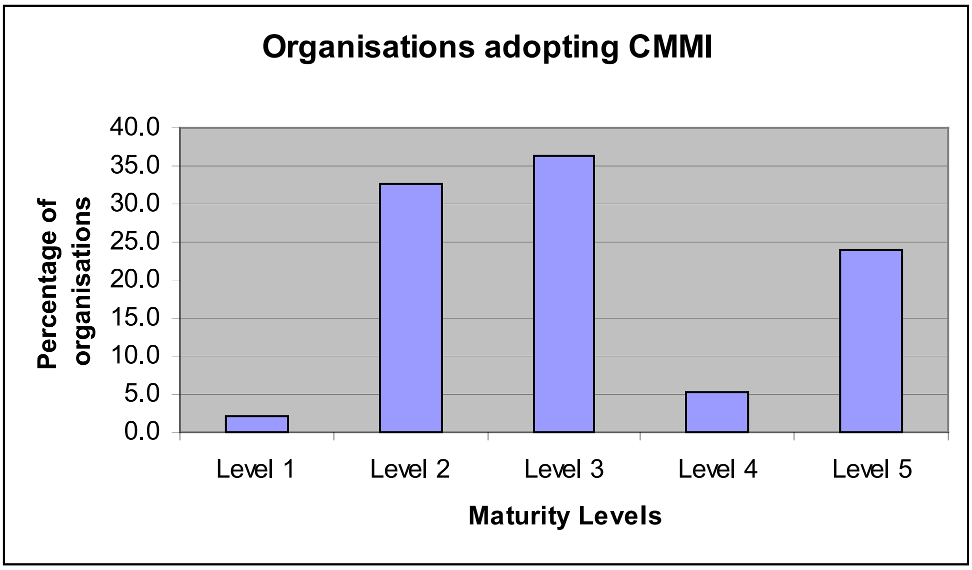


Figure C.7: Organisations distributed by maturity levels (source data: Software Engineering Institute, (2005))

7. ISO/IEC Software Process Assessment

In June 1995, the SPICE Project Organisation put forward a set of working drafts among which is a model for process management (SPICE, 1995). The model has six maturity levels as follows:

Level 0 - Not-Performed

There is general failure to perform the base practices in the process. There are no easily identifiable work products or outputs of the process.

Level 1 - Performed-Informally

Base practices of the process are generally performed. The performance of these base practices may not be rigorously planned and tracked. Performance depends on individual knowledge and effort. Work products of the process testify to the performance. Individuals within the organisation recognise that an action should be performed, and there is general agreement that this action is performed as and when required. There are identifiable work products for the process.

Level 2 - Planned-and-Tracked

Performance of the base practices in the process is planned and tracked. Performance according to specified procedures is verified. Work products conform to specified standards and requirements. The primary distinction from the Performed-Informally Level is that the performance of the process is planned and managed and progressing towards a well-defined process.

Level 3 - Well-Defined

Base practices are performed according to a well-defined process using approved, tailored versions of standard, documented processes. The primary distinction from the Planned-and-Tracked Level is that the process of the Well-

Defined Level is planned and managed using an organisation-wide standard process.

Level 4 - Quantitatively-Controlled

Detailed measures of performance are collected and analysed. This leads to a quantitative understanding of process capability and an improved ability to predict performance. Performance is objectively managed. The quality of work products is quantitatively known. The primary distinction from the Well-Defined Level is that the defined process is quantitatively understood and controlled.

Level 5 - Continuously-Improving

Quantitative process effectiveness and efficiency goals (targets) for performance are established, based on the business goals of the organisation. Continuous process improvement against these goals is enabled by quantitative feedback from performing the defined processes and from piloting innovative ideas and technologies.

The primary distinction from the Quantitatively-Controlled Level is that the defined process and the standard process undergo continuous refinement and improvement, based on a quantitative understanding of the impact of changes to these processes.

This model is similar to the SEI's Capability Maturity Model, except that it has a level 0.

Shortcomings:

(i) Political, socio-economic and cultural aspects are not considered.
(ii) Soft skills are not considered.

APPENDIX D

Details on Continuous Process Improvement for Case study 1

Continuous Process Improvement group of KPAs

Change Management

Projects are often affected by a change in technology, user resistance and changes in requirements. Therefore, changes need to be managed effectively and efficiently.

Using the ESPM[3], change management was carried out using the following inputs:

- Nature of change (whether affecting organisation, technology, staff, etc.).
- Historical data.
- Change management action knowledge base.

The following outputs were considered:

- Successful management of changes.
- Improved organisation efficiency.
- Lessons learnt for future improvement of change management.

The transformation of the inputs to the outputs is shown in figure D.1.

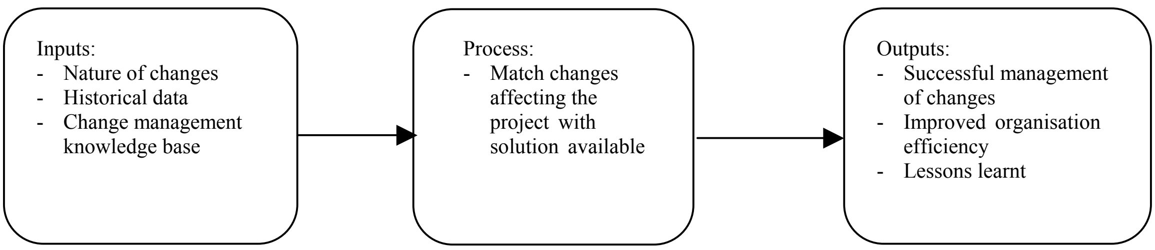


Figure D.1: Change management – Transformation of inputs into outputs

Change management was mapped onto four process groups as per the PDCA cycle shown in figure D.2.

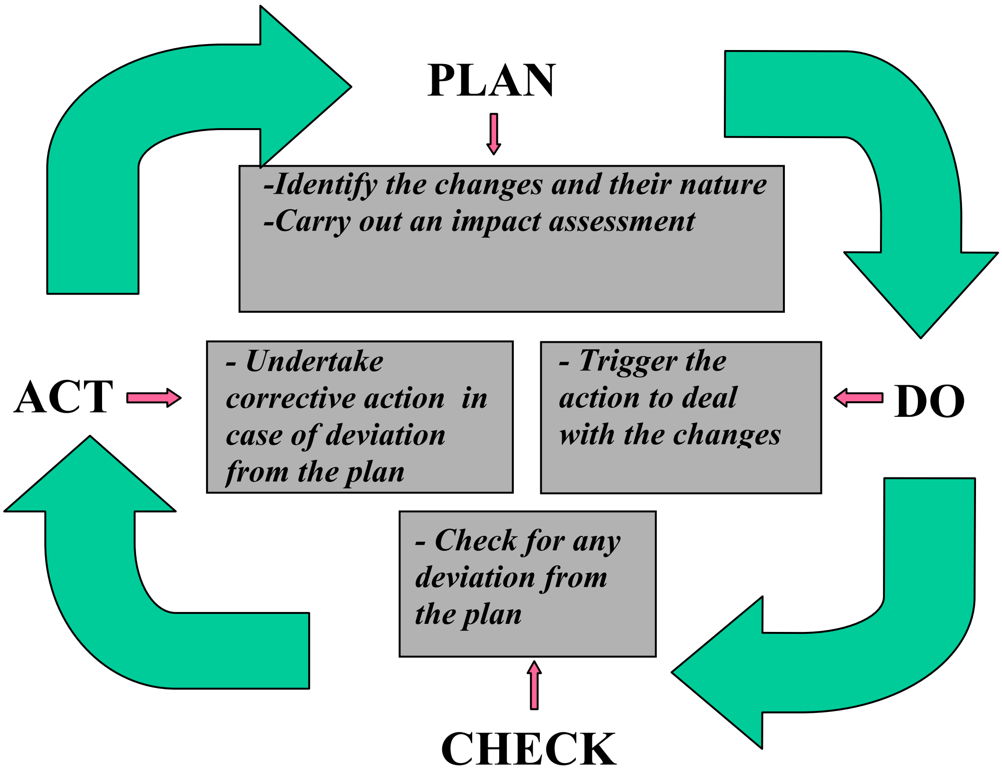


Figure D.2: PDCA cycle for change management

Plan: Change Management

Planning was carried out for changes in technology, user resistance and changes in requirements. The staff members of the organisations were exposed to new technologies in the areas of hardware, software tools and telecommunications through regular trainings. They were, therefore, well informed about the new technologies and they were ready to shift to new technologies as and when they were proved to be well established.

It was believed that the importance of the project would discourage user resistance. Regular weekly meetings were also considered to help in resolving any problems of user resistance should it occur during any stage of the project. Furthermore, the inclusion of the experienced users during all stages of the project would facilitate a smooth development of the project and also allow users to be familiar with the forthcoming changes that would affect their way of carrying out their work.

The requirements were gathered during the first activity (Analysis). The functionality of the software was determined during that stage and upon validation and approval of management, the software would be developed. Any major change would be considered as a separate assignment and would be governed by a new contract.

Do: Change Management

Minor changes in the functionality of the project were noted and these could be accommodated easily within the time schedule prepared.

No change in technology affected the project and user commitment was ensured through the assignment.

Check: Change Management

Although some minor changes to the requirements were noted, no changes in time schedules were necessary for this project.

Act: Change Management

Only minor changes were necessary for the project. Therefore, action was taken to incorporate these changes for the project.

Software specific focus

Given the specificity of the project (namely a software project), the associated strengths and weaknesses were considered. In this context, advantage was taken of the strengths while at the same time care was taken to deal with the weaknesses.

The strengths of the project were considered to be its:

- Flexibility.
- Mobility.
- Replication.
- Scalability.
- Backup.
- Reusability of components.
- Possibility of using prototypes.
- Software evolution.

The weaknesses of the project were considered to be its:

- Invisibility.
- Complexity.
- Difficulty regarding estimation.
- Dynamic nature.
- Intangibility.
- Regular upgrades.
- Bugs.
- Difficulty to add people to a delayed project.
- Training.

The PDCA cycle for the Software specific focus

The following inputs were considered:

- Identified strengths and weaknesses.
- Appropriate actions to deal with corresponding strengths and weaknesses.
- Historical data.

The outputs of the software specific process were:

- Specific management techniques for software projects.
- Improved productivity.

The transformation of the inputs to the outputs is shown in figure D.3.

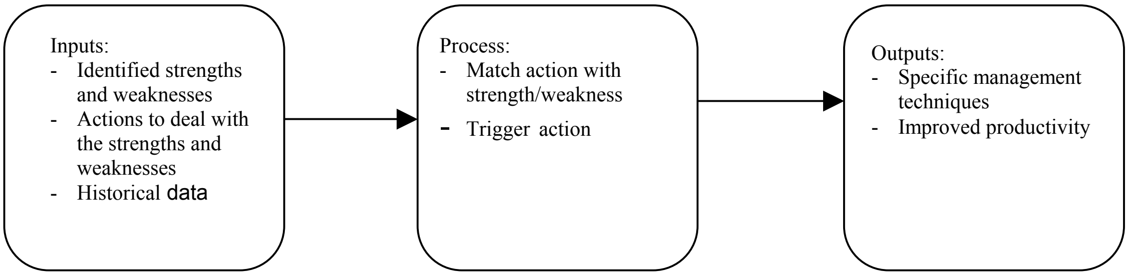


Figure D.3: Software specific focus - Transformation of inputs into outputs

The strengths and weaknesses were mapped onto four process groups as per the PDCA cycle shown in figure D.4.

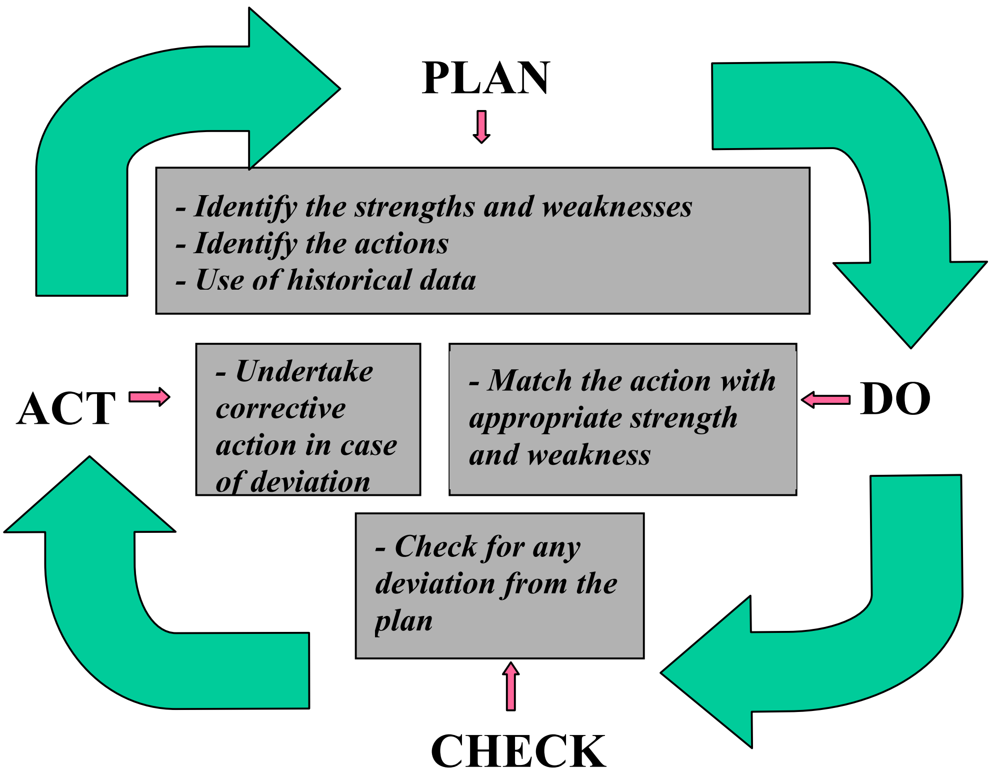


Figure D.4: PDCA cycle for software specific focus

Plan: Software specific focus

The specificities of the project based on the strengths and weaknesses identified above were considered in the planning process group. The planning process group for the strengths took care of the advantages and disadvantages discussed below.

The strengths considered for the project were:

- *Flexibility*. Flexibility was facilitated by the modular design approach. Therefore, any new functionality could be incorporated without disrupting the design of the software.
- *Replication*. Replication was not a concern in this project, but it would have been an advantage in case the supplier would be required to develop a similar software for another organisation.
- *Scalability*. The scalability feature could be associated with the flexibility strength whereby the new modules could be added.
- *Backup*. This feature ensured an easy and rapid transfer between the contractor's site and the client's site.
- *Reusability of components*. It was planned to use as many components as were available. Screens for login, menu design, code for encryption of password, procedures for audit trail, code for push buttons and validation were planned for use.
- *Software evolution*. It was planned to make use of any new technology as regards to software and hardware. Staff members of the software development company were exposed to new technologies through regular training sessions. The software company applies the latest, established technology in its projects.

Similarly, the weaknesses were taken care as follows:

- *Invisibility*. It was planned to render visible as many activities as possible. In this context, the planned and actual activities were represented in a Gantt chart.
- *Complexity*. Complexity was reduced by using a Work Breakdown Structures (WBSs),
- *Difficulty regarding estimation*. Estimation was based on object point analysis and estimation by analogy.
- *Dynamic nature*. The latest established technology used facilitated a longer life span of the software.
- *Intangibility*. User involvement was encouraged throughout the development of the project due to the inherent intangibility of software.
- *Regular upgrades*. Regular upgrades are major obstacles for the developer as it requires the recompilation of some components and the testing and replacement of the hardware in some cases. With the use of the most recent established technology, a long life span was facilitated before any further upgrades would become necessary.
- *Bugs*. Bugs are major obstacles that developers have to overcome before the software is accepted. It was planned to have user involvement during all stages of the project and quality control checks after the development of each module in order to prevent failure of the product.
- *Difficulty to add people to a delayed project*. It was planned not to add staff near the end of the project.
- *Training*. Training was planned for all officers who would use the system.

Do: Software specific focus

The planned tasks to deal with the strengths and weaknesses were executed. Actions were taken to make use of the strengths and counteract the weaknesses as discussed next.

The strengths that were considered were:

- *Flexibility*. Modules were defined following a Work Breakdown Structure. The coding of the modules were undertaken and integrated to build the final product.
- *Replication*. Replication was not relevant for the project.
- *Scalability*. The modular approach used facilitated the adding of new functionality in the future.
- *Backup*. This feature was used to replicate the software at the contractor's site for training purposes in the event that the client's site was not ready.
- *Reusability of components*. Screens for login, menu design, code for encryption of passwords, procedures for audit trails, code for push buttons and validation were used to save time during the development phase.
- *Software evolution*. Although there was no impact of changing technology, staff members of the software development organisation were constantly kept aware of new technologies. The modular structure of the software ensured that the software would adapt to changes in business rules more easily.

Similarly, the weaknesses that were considered were:

- *Invisibility*. The use of a Gantt chart and software tools (namely Microsoft Project) rendered many activities of the project visible.
- *Complexity*. Complexity was reduced by making use of a Work Breakdown Structure.
- *Difficulty regarding estimation*. Estimation was carried out using object point analysis.
- *Dynamic nature*. The latest established technology was used so that the software could be used for a longer time before an upgrade would be required.

- *Intangibility*. User involvement was considered throughout the project to ensure that users were aware about the matching of the requirements and the functionality of the software.
- *Regular upgrades*. The use of the new established technology ensured a long life span for the software product before any further upgrade.
- *Bugs*. Most bugs were identified during the programming activity with the help of the users and they were immediately corrected.
- *Difficulty to add people to a delayed project*. There was no need to add more people on the project.
- *Training*. Training was conducted at the user's site.

Check: Software specific focus

The action undertaken during the previous process group was monitored in order to detect any deviation from the planned process group.

The strengths were:

- *Flexibility*. The modules were defined, developed and integrated following a Work Breakdown Structure as planned.
- *Replication*. This strength was not of major concern in the project although software could be replicated if there was a need.
- *Scalability*. The modular approach was adopted as planned.
- *Backup*. Regularly, a backup of the software was kept in order not to lose the software in the case of hardware or software failure.
- *Reusability of components*. As far as possible, screens for login, menu design, code for encryption of passwords, procedures for audit trails, code for push buttons and validation were reused.
- *Software evolution*. As planned, the need for the software to adapt to changing business rules was facilitated through a modular development approach.

Similarly, the weaknesses considered were:

- *Invisibility*. The visibility of activities involved in the project was rendered through the use of a Gantt chart and software tool (namely Microsoft Project).
- *Complexity*. It could be noted that complexity of the software was actually reduced through the use of a Work Breakdown Structure.
- *Difficulty regarding estimation*. Object point analysis and estimation by analogy helped in rapidly estimating the cost of the software. The software developer also used that method in cost estimation. Therefore, there was no difficulty in using such an estimation technique.
- *Dynamic nature*. Given the nature of a software project, whereby the software had to be modified to adapt to changing business rules, minimum difficulties were encountered due to the modular approach adopted.
- *Intangibility*. The intangibility weakness was overcomed by allowing the users to be involved throughout the development of the project.
- *Regular upgrades*. The latest established technology was used to facilitate a long life span of the software product.
- *Bugs*. Bugs were identified during the programming stage and they were removed immediately. Additional testing was carried out by the testing team.
- *Difficulty to add people to a delayed project*. There was no need to add more people to the project.
- *Training*. During the check process group, training was found to be as per a revised plan.

Act: Software specific focus

This process group took into consideration the output of the previous process group (Check process group) as a feedback to determine any corrective action deemed necessary. Any deviation that occurred was rectified as soon as possible.

No deviation in the plan was noted with respect to flexibility, replication, scalability, backup, reusability of components and software evolution. In addition, no deviation was also encountered with respect to invisibility, complexity, difficulty regarding estimation, dynamic nature, intangibility, regular upgrades, bugs, difficulty to add people to a delayed project, and training (although the exercise was conducted at the contractor's site).

Environmental management

The social, economic and political aspects were relevant for this project.

The political dimension refers to the political benefits to be derived by developing a citizen-centric application. Citizens would be able to find vacancies from a website and register as job seekers and provide information on themselves so as to allow the system to match their profiles against selection criteria for specific jobs.

As discussed in chapter 3, Mauritius being an African developing country is affected by high power distance and moderate uncertainty avoidance. These factors were taken into consideration during the management of the project. The project manager had to intervene to counteract the effect of high power distance. Similarly, moderate uncertainty avoidance was considered to avoid failure of the project. Collectivism and masculinity/femininity dimensions were not found to be of importance in this project.

The PDCA cycle for Environmental management

In managing the environmental factors, the following inputs were transformed into outputs:

- List of environmental factors to be addressed.
- List of ways to deal with the factors.
- Historical data.

The transformation generated the following outputs:

- Satisfied team members.
- Safe software development environment.
- Lessons learnt.

The transformation of the inputs to the outputs is shown in figure D.5.

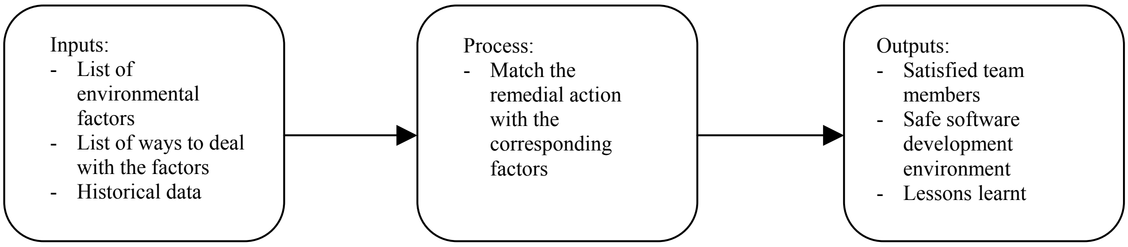


Figure D.5: Environmental management - Transformation of inputs into outputs

In the ESPM[3,] the environmental factors are taken into consideration. For this project, a list of factors was prepared and actions to be taken were defined. These factors were mapped onto four process groups as per the PDCA cycle shown in figure D.6.

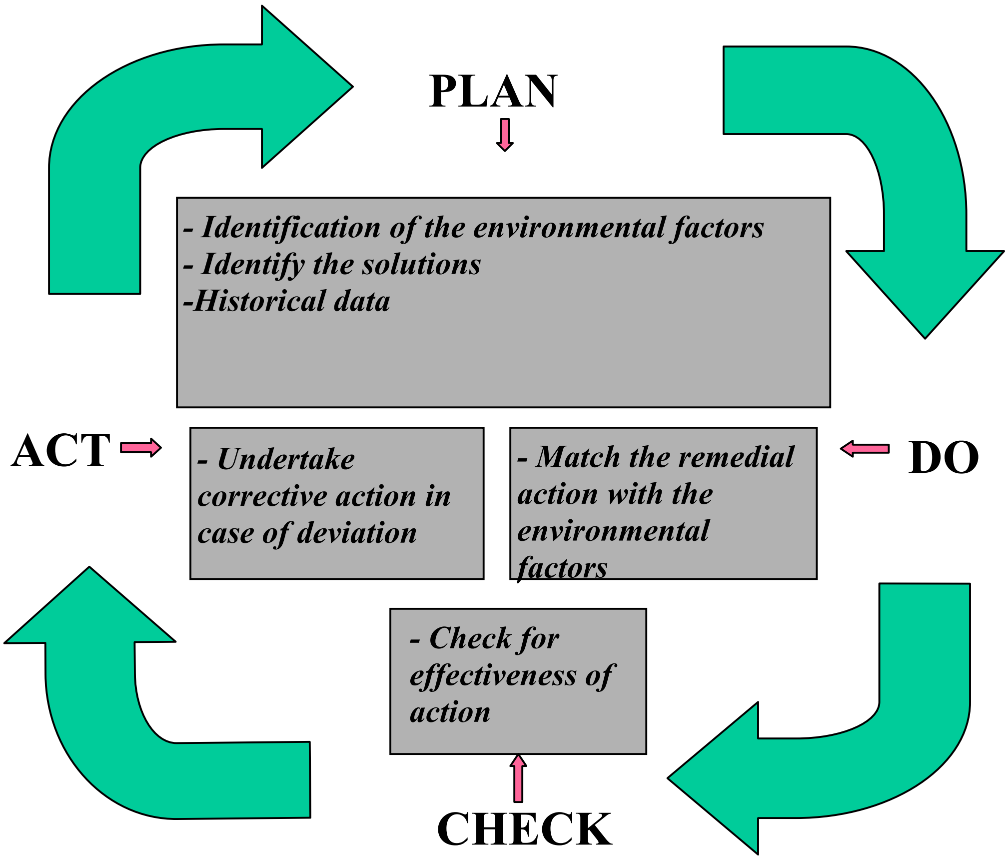


Figure D.6: PDCA cycle for Environmental management

Plan: Environmental management

The relevant factors were listed for consideration during the project execution. The social, economic and political dimensions were identified for the project as it could affect the society by availing them of more free time and at the same time the political objective would be met by providing services online to the public (citizen-centric applications).

In addition, high power distance was assumed to be of concern both at the software development organisation as well as at the client's organisation. Communication between top management and the software developers was rare. It was planned that the project manager would act as the link between top management and the software developers.
As the project was taken to be a priority one and it was expected to provide political, social and economic benefits, risks were avoided as far as possible. As such, ambiguity perceived as threatening and risk-taking behaviour were

avoided by planning for an outsourced solution. The software development company also avoided uncertainty by using a development platform and technology with which the development team was familiar.

Do: Environmental management

The plan for environmental management included the social, economic and political dimensions.

Social benefits were ensured by implementing a solution so that citizens could view vacancies from a website round the clock. At the same time, the efficiency of operations of that department would be enhanced through the back office software that would undertake the matching of job seekers against the vacancies available and notify employers accordingly.

In order to meet the economic objectives, the system would also record the skills available in the country and the skills and profiles of candidates required. This would help the authority concerned to plan for skills required in the future so that the country would have skilled labour in various fields so as to sustain the economy.

In addition, the political benefits of providing services online would be achieved. The advertisement of this new channel of dissemination of information on vacancies and online registration of job seekers would bring about political benefits as well.

During the execution of the project, the project manager handled issues like leave and irregular working hours. He had to allow the staff to work with enough flexibility so as to ensure the necessary level of motivation. He had to resolve all problems faced by the development team.

At the client's organisation, there was the option of carrying out the development of the software in-house through the services of another department. As a result of the importance of the project, it was decided to seek the services of an external firm that was specialised in the development of such software. A tender exercise was carried out to select companies with the required experience to provide the hardware as well as develop the software and website. The software development organisation made use of a development platform and technology that was well established. Although the development staff member was regularly exposed to new technology through training programmes, it was not intended to explore new avenues for such an important project.

Check: Environmental management

Regularly, the social, economic and political objectives planned had to be checked. Much attention was paid to the functionality of the software to be citizen-centric while at the same time enhancing the operations of the client's organisation.

As far as power distance was concerned, it was found that the project manager managed to keep all issues raised by the development team under control. These issues pertained mainly to leave and flexibility in working hours. As the duration of the project was quite long, it was fair to provide such flexibilities in order to keep the staff motivated.

The decision to outsource the software development task proved to be a wise one as most of the activities were completed on schedule. The users were satisfied with the work of the development team. The experience of the development team and their commitment was well acknowledged by the users. In this way, uncertainty was avoided by the users rather than seeking in-house assistance for such a specialised development exercise. In-house development would have incorporated risks of timely delivery and low quality of the product, thereby compromising the success of the project. The development team also avoided any uncertain outcomes by making use of well-established development platform and technology.

Act: Environmental management

The website was developed with the citizen's needs in mind in order to bring about social and political benefits. The back office application ensured the economic benefits by processing the data received efficiently.

The high power distance prevailing in Mauritian organisations was an issue that had to be resolved by the project manager. This not only ensured the motivation within the development team but also contributed towards establishing a conducive environment. The project manager had to be on his guard to be able to allow a trust relationship in order to know about the problems of the staff and to resolve these, as and when they arose.

The fact that uncertainties were avoided in this project contributed to a success of the project (at least for the agreed time, cost and quality) although better results could have been obtained. In the case of such a project, uncertainty avoidance was found to be less risky. Unfortunately, moderate uncertainty avoidance for low priority projects may deprive them from reaching better results in terms of time, cost and quality.

Integration management

Integration testing was carried out by execution of the PDCA cycle so that the KPAs could be dealt with in a coordinated manner with the ultimate goal of achieving the overall objective of the project. In the ESPM[3], the integration management KPA ties all the other KPAs together so as to ensure that a common goal is attained.

The following inputs were considered in integration management:

- Other KPAs.
- Constraints.

The outputs generated were:

- Project results.
- Lessons learnt.

The transformation of the inputs to the outputs is shown in figure D.7.

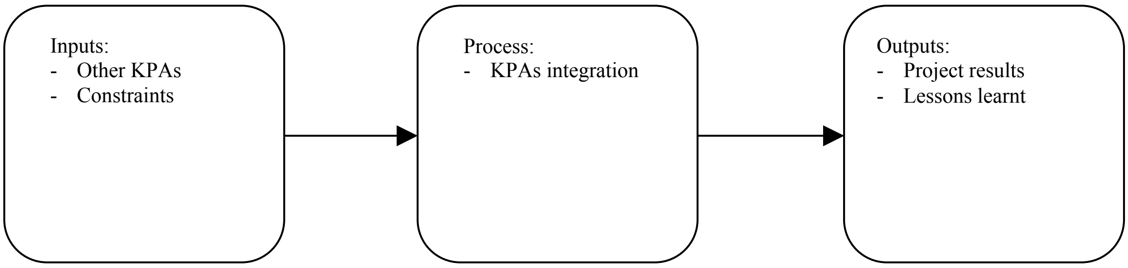


Figure D.7: Integration management - Transformation of inputs into outputs

The mapping of integration management onto the four process groups as per the PDCA cycle shown in figure D.8.

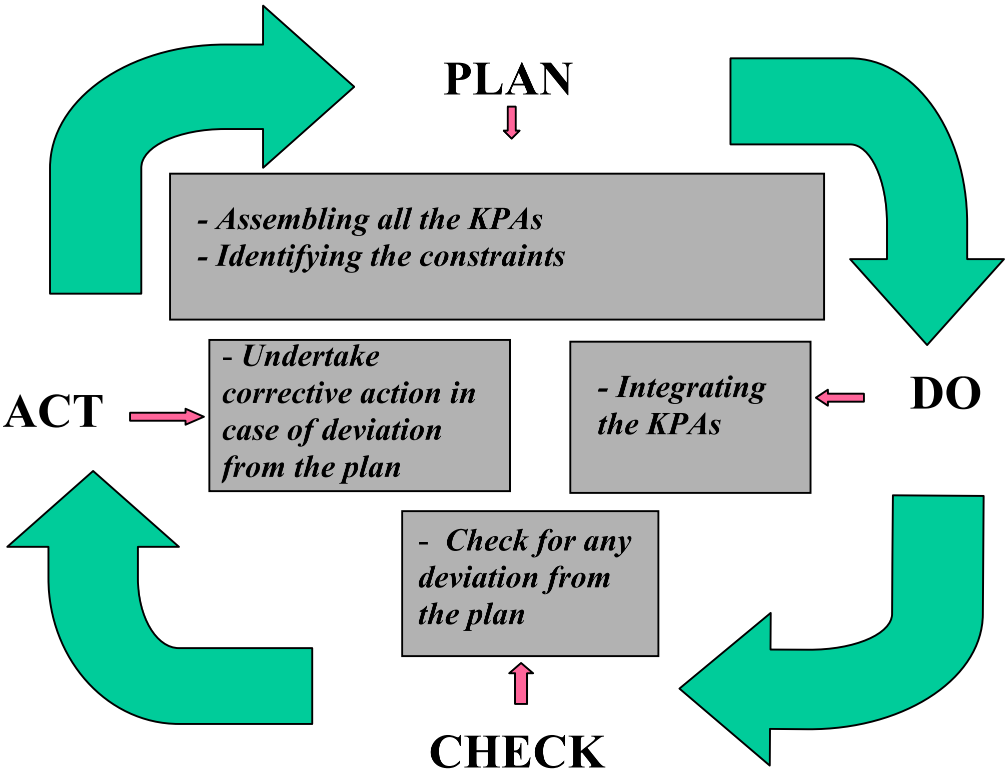


Figure D.8: PDCA cycle for Integration management

Plan: Integration management

The objective of the planning process group for integration management was to bind all KPAs together so that they could be dealt with in a coherent manner. The project manager had to keep track of all the KPAs to enhance the success of the project (see table 12.8).

Table D.1: Integration management planning

Continuous Process Improvement Group		**Level 2 KPAs**
Integration Management	Soft skills management	Time management
		Cost management
	Change management	Quality management
		Level 3 KPAs
	Software specific focus	Human resource management
	Environmental management	Risk management
		Contract management

The planning process group for each KPA was defined in parallel. All the plans were prepared for simultaneous execution. The details were described above for the respective KPAs.

Do: Integration management

This process group is concerned with the execution of the plans for all the KPAs. All the plans from the previous process groups were executed in a coordinated manner. With the overall objective of successful project delivery, each of the “Do” process groups was executed and the next process group was executed. The results of the “Do” process group for all the KPAs were entered into the Microsoft Project software and displayed on the Gantt chart.

Check: Integration management

The actual results were compared with the planned results in a coordinated manner. The details for each KPA have been explained before. The planned and actual progress of the activities and the resource usage were monitored.

Act: Integration management

After the comparison of the actual results against the planned results, any deviation was identified and the necessary actions were taken. It could be noted that the project was executed successfully as planned.

APPENDIX E

Details on Continuous Process Improvement for Case study 2

Continuous Process Improvement group of KPAs

Change Management

Change in technology, construction of a new computer room followed by the relocation of the computer equipment, user resistance to carry out the acceptance test at the software company's site as well as replacement of one key staff member occurred during the project execution. Not all of these events caused delay in the project completion.

A change in technology was brought by the introduction of a new version of the Database Management System. The user decided to opt for the new version of the DBMS as support would be available for a longer time. This is due to the fact that after some years, an old version of the DBMS would not be supported. The new version would ensure a longer life span before an upgrade would be required. The software development organisation agreed to the incorporation of the new technology. Unfortunately, the programmer had to spend more time to become familiar with the new version of the DBMS. This led to a delay of four person-months as regard to the programming activity. Therefore, this change in technology caused a review of the time schedule. The time management activity required an iteration of the PDCA cycle.

The construction of the new building with a computer room resulted in a delay and user reluctance was encountered for the acceptance test to be conducted at the software development organisation's site. The project manager managed to conduct the training at the software development organisation's site as there was major disruption due to construction works at the client's site.

The loss of one key staff member before the completion of the project was proactively handled by the change management process group.

Using the ESPM[3], change management was carried out using the following inputs:

- Nature of change (whether affecting the organisation, technology, staff, etc.).
- Historical data.
- Change management action knowledge base.

The following outputs were considered:

- Successful management of changes.
- Improved organisational efficiency.
- Lessons learnt for future improvement of change management.

The transformation of the change management inputs to the outputs is shown in figure E.1.

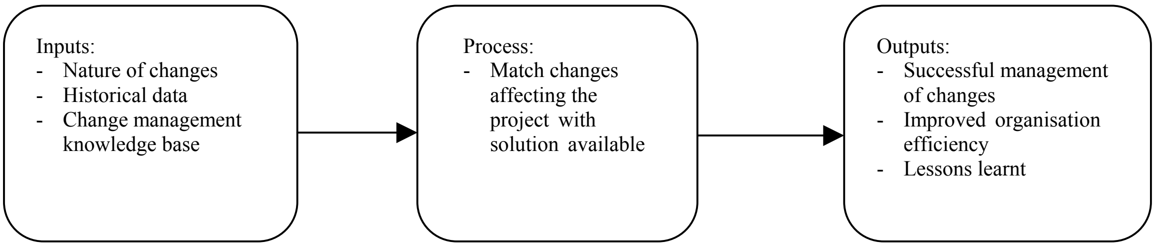


Figure E.1: Change management - Transformation of inputs into outputs

Change management was mapped onto four process groups as per the PDCA cycle shown in figure E.2.

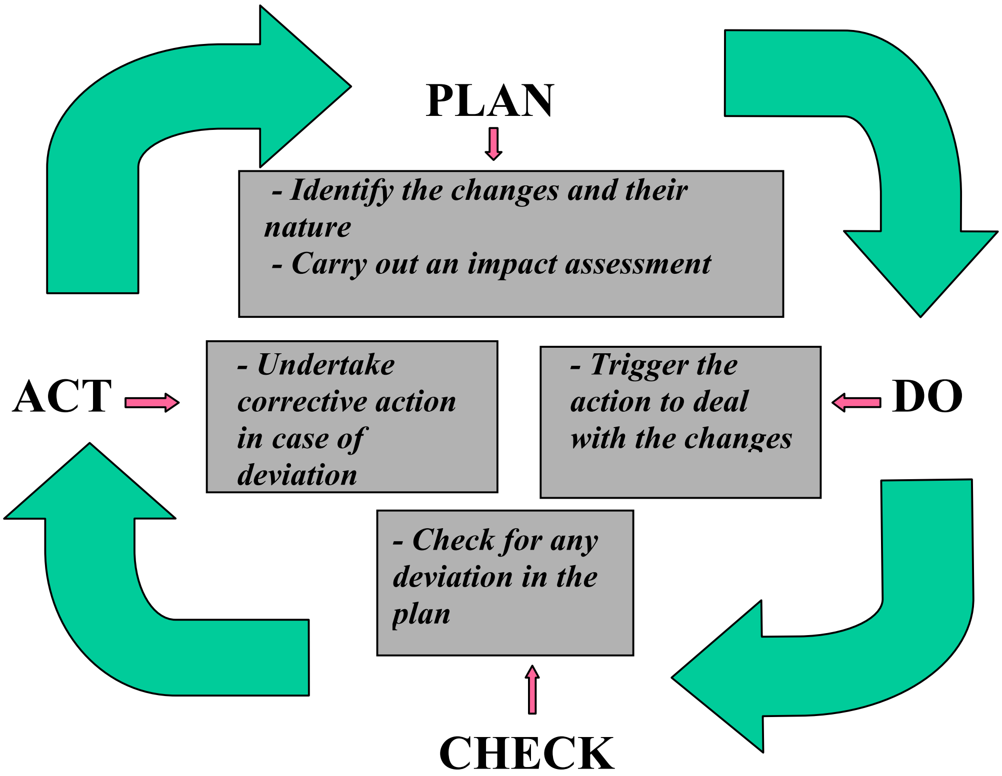


Figure E.2: PDCA cycle for change management

Plan: Change Management

During the planning process group for change management, the impact of a change in technology was considered with a view to minimise delays arising from the introduction of new technology to which the staff of the contractor's organisation have not been exposed before. Due to the fact that the user insisted on the use of the new technology while the project was already in progress, the software developer agreed to incorporate the new version of the DBMS but at the expense of a delay in the project. No additional cost was charged as the software developer could recoup that amount from future projects. The experience with the new version of the DBMS would be obtained from the current project and this knowledge would be used in future projects.

The loss of a key staff member during the execution of the project was planned such that should an event occur, a replacement staff member could take over in the minimum amount of time. The existing model (CMMI) that was in use and known to all the staff contributed to the smooth replacement of the staff.

Changes were brought about in the planned activities by the relocation of the computer equipment. This caused delays that could not be resolved due to user resistance for acceptance testing at the contractor's site.

Do: Change Management

Changes in the new DBMS technology were incorporated so that the user could benefit from an improved version of the software. The time for the completion of the project did not include slack time to accommodate for changes in technology. Therefore, the use of the new version of the DBMS was made by an extension in the programming activity.

The loss of any staff members during the execution of the project was foreseen as a likely event. Therefore, the planning of this event allowed the replacement of the programming staff during the acceptance phase. No major disruption occurred although the major delay took place due to user reluctance to undertake the activity.

The relocation of the computer room was not foreseen. However, the contractor had an option available in order to minimise delays (namely training and acceptance test at the contractor's site). Training was dispensed at the contractor's site while provision was made for acceptance test. Unnecessary delay occurred due to user reluctance. This could have been prevented in the event that the required user commitment was available.

Check: Change Management

The results obtained during the "Do" process group were monitored and analysed during the check process group. Technology change, namely with respect to the new DBMS, indicated an increase in the time taken to complete the programming activity. The relocation of the computer room brought major delays that impacted on the schedule of all the activities.

Fortunately, the loss of one team member was well handled and there was no impact on the schedule. The team member was readily replaced by another experienced staff member. An extension of a further three months was required to complete the project after the user agreed to start with the acceptance phase. The contractor then started to remove all the bugs that were identified during the acceptance test phase.

Act: Change Management

Action was taken to rectify problems encountered during the "Check" process group. Necessary changes were brought to the project schedule produced due to changes in technology. The delay in the acceptance test phase was attributed partly to the relocation of the computer room and the reluctance of the user to undertake the activity at the software development company's site. This prompted the company to request for its payment due to non-compliance of the client to the contract. This action on the part of the contractor caused weekly meetings to be held instead of monthly meetings in order to complete the project according to specifications (required quality) although it was late and the contractor did not request for additional payment.

The loss of one team member during the acceptance phase was noted during the "Check" process group and action was immediately taken to replace him.

Software specific focus

Given the specificity of the project (namely a software project), it was worth handling the strengths and weaknesses associated with it. In this context, advantage was taken of the strengths while at the same time care was taken to deal with the weaknesses.

The strengths considered for the project were:

- Flexibility.

- Mobility.
- Replication.
- Scalability.
- Possibility of using backup.
- Reusability of components.
- Possibility of using prototypes.
- Software evolution.

The weaknesses considered for the project were:

- Invisibility.
- Complexity.
- Difficulty regarding estimation.
- Dynamic nature.
- Intangibility.
- Regular upgrades.
- Bugs (inherent in software).
- Difficulty to add people to a delayed project.
- Training.

PDCA cycle for Software specific focus

The following inputs were considered:

- Identified strengths and weaknesses.
- Appropriate actions to deal with corresponding strengths and weaknesses.
- Historical data.

The outputs of the process generated were:

- Specific management techniques for software projects.
- Improved productivity.

The transformation of the software specific inputs to the outputs is shown in figure E.3.

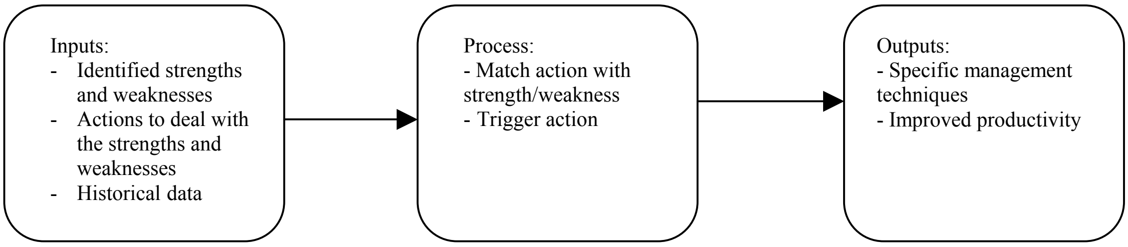


Figure E.3: Software specific focus - Transformation of inputs into outputs

The strengths and weaknesses were mapped onto four process groups as per the PDCA cycle shown in figure E.4.

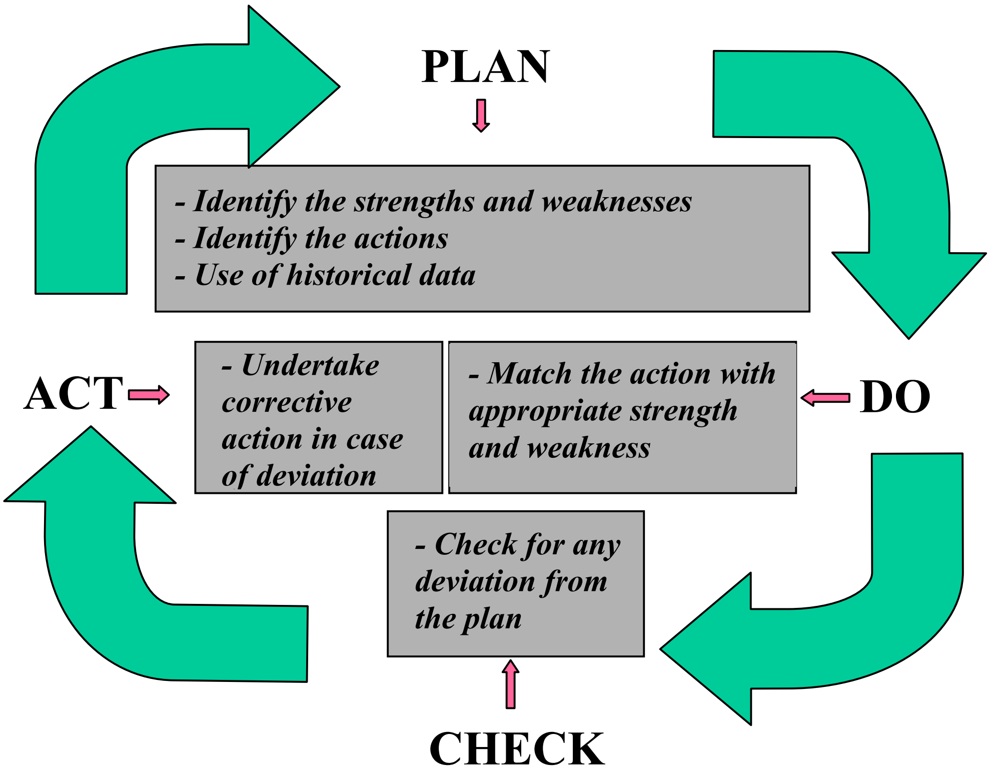


Figure E.4: PDCA cycle for software specific focus

Plan: Software specific focus

The specificities of the project based on the strengths and weaknesses identified above were considered in the planning process group. The planning process group for the strengths took care of the following:

The strengths were taken care of as follows:

- *Flexibility*. Flexibility was ensured by the modular design approach. Therefore, any new functionality could be incorporated without disrupting the entire design of the software.
- *Mobility*. The mobility of software was not of concern for this project as it was important to be copied at the software development company's site for training purposes.
- *Replication*. It was planned to make use of the same software with certain customisations at another department.
- *Scalability*. The scalability feature could be associated with the flexibility strength whereby the new modules could be added.
- *Backup*. This feature ensured easy and rapid transfer between the contractor's site and the client's site.
- *Reusability of components*. It was planned to use as many components available. Thus, screen for login, menu design, code for encryption of password, procedures for audit trail, code for push buttons and validation were planned for use.
- *Possibility of using prototypes*. A prototype was planned for development in order to clear up possible misunderstandings between the client and the developer.
- *Software evolution*. It was planned to make use of any new technology with regards to software and hardware. A hardware and software research & development team at the software company keep track of developments in the field. Beta versions of software is normally tested and information on new developments in the hardware arena is communicated to all the staff of the organisation.

Similarly, the weaknesses were taken care of as follows:

- *Invisibility*. It was planned to render visible as many activities as possible. In this context, the planned and actual activities were represented in a Gantt chart.
- *Complexity*. Complexity was reduced by planning for Work Breakdown Structure (WBS). Gantt chart is the most commonly used project management tool in Mauritius (Sukhoo et al., 2004b) and this tool was used.
- *Difficulty regarding estimation*. Estimation was based on object point analysis.
- *Dynamic nature*. The latest version of the DBMS facilitated a longer life span of the software.
- *Intangibility*. Given the intangibility of the product, it was decided to make use of a prototype to demonstrate the functionalities of the product.
- *Regular upgrades*. Regular upgrade is a hassle for the developer as it requires recompilation of some components, testing and replacement of the hardware in some cases. With the use of the most recent version of the DBMS, a long life span was facilitated before any further upgrade.
- *Bugs*. Bugs are major obstacles that developers have to overcome before software is accepted. It was planned to have quality control checks after the development of each module in order to prevent failure of the product.
- *Difficulty to add people to a delayed project*. It was planned not to add staff near the end of the project.
- *Training*. Training was planned for all officers who would use the system.

Do: Software specific focus

The planned tasks to deal with the strengths and weaknesses were put into action. Steps were taken to utilise the strengths and counteract the weaknesses as follows:

The strengths were taken care of as follows:

- *Flexibility*. Modules were defined following a Work Breakdown Structure. The coding of the modules were undertaken and integrated in order to build the final product.
- *Mobility*. Given that this was not an issue for this project, this inherent strength was ignored.
- *Replication*. The requirements for another similar software was gathered at the same time.
- *Scalability*. The modular approach used ensured that new functionalities could be added in the future.
- *Backup*. This feature was used in order to replicate the software at the contractor's site for training purposes.
- *Reusability of components*. Screens for login, menu design, code for encryption of passwords, procedures for audit trails, code for push buttons and validation were used to save time during the development phase.
- *Possibility of using prototypes*. A prototype was built to show the user interface as well as certain functionalities. This helped the user to have a feel of what the final product would look like.
- *Software evolution*. Experience was acquired in development using the latest version of the DBMS. This would help in the development of other systems taking advantage of the new technology.

Similarly, the weaknesses that were taken care of were:

- *Invisibility*. The use of a Gantt chart and software tool (namely Microsoft Project) rendered many activities of the project visible.
- *Complexity*. Complexity was reduced by making use of a Work Breakdown Structure.
- *Difficulty regarding estimation*. Estimation was carried out using object point analysis.

- *Dynamic nature*. The latest version of the DBMS was used so that the software could be used for a longer time before an upgrade would be required.
- *Intangibility*. The use of a prototype before the actual development of the system ensured that the user obtained a feel of what the final product would look like.
- *Regular upgrades*. A software product suffers from the need for regular upgrades in case a new version of a DBMS becomes available. In the case of the project, the change to the new version of the DBMS will ensure that the software will have a longer life span. An old version of the DBMS may cause support to be discontinued for the product in the short run.
- *Bugs*. Many bugs were detected during the acceptance testing phase. The project had to be extended for a further period of 3 months in order to correct the bugs.
- *Difficulty to add people to a delayed project*. There was no need to add more people to the project.
- *Training*. Training was conducted at the contractor's site as the user's site was not ready.

Check: Software specific focus

The actions undertaken during the previous process group was monitored in order to detect any deviation from the planned process group.

The strengths were taken care of as follows:

- *Flexibility*. The modules were defined, developed and integrated following a Work Breakdown Structure as planned.
- *Mobility*. Mobility of software was considered by installing the software at the premises of the software development company to conduct training.
- *Replication*. The software was built in view of replicating the software with minimum customisation.

- *Scalability*. The modular approach was adopted as planned.
- *Backup*. In order to conduct training, a copy of the software was also installed at the contractor's site. The contractor was also eager to use the copy for acceptance tests, but due to the reluctance of the user, this activity could not take place at that site.
- *Reusability of components*. As far as possible, screens for login, menu designs, codes for encryption of password, procedures for audit trail, codes for push buttons and validation routines were used again. However, some components had to be recompiled using the new version of the DBMS.
- *Possibility of using prototypes*. It is noted that a prototype was used so as to clear up any initial misunderstandings on the part of the user.
- *Software evolution*. As planned, experience was obtained in using the last version of the DBMS and advantage was taken of the new technology available. In addition, recompilation of certain components ensured their availability for new systems and the upgrading of other existing systems.

Similarly, the weaknesses were taken care of as follows:

- *Invisibility*. The visibility of activities involved in the project was rendered through the use of a Gantt chart and software tool (namely Microsoft Project).
- *Complexity*. A WBS was used to reduce the complexity of the software.
- *Difficulty regarding estimation*. Object point analysis helped in rapidly estimating the cost of the software. The software developer was also making use of that method in cost estimation. Therefore, there was no difficulty in using such an estimation technique.
- *Dynamic nature*. Given the nature of a software project, whereby software has to be modified in order to adapt to changing business rules, the modular approach ensured that this would be possible with minimum difficulties.

- *Intangibility*. The intangibility weakness was addressed by allowing the user to have a feel of the software prior to the actual development through the use of a prototype.
- *Regular upgrades*. The latest version of the DBMS was used to facilitate a long life span before a newer version of the DBMS become available.
- *Bugs*. Many bugs were detected during the acceptance testing phase. The contractor gave the commitment that all the bugs would be removed during a period of three months at no additional cost.
- *Difficulty to add people to a delayed project*. There was no need to add more people to the project.
- *Training*. During the check process group, training was found to be as per a revised plan.

Act: Software specific focus

This process group took into consideration the output of the previous process group (Check process group) as a feedback in order to determine any corrective action deemed necessary. Any deviation that occurred was rectified as soon as possible.

The strengths were taken care of as follows:

- No deviation from the plan was noted in the case of *flexibility, mobility, replication, scalability, reusability of components, possibility of using prototypes* and *software evolution*.
- *Backup*. In the case of acceptance test, the reluctance of the user to carry out the activity at the contractor's site caused a major delay in delivery of the software. The project manager had to intervene by asking Management of the client's organisation to provide the necessary support. This could only be achieved by claiming for payment due to a delay in conducting the acceptance test by the user.

The weaknesses were taken care of as follows:

- No deviation from the plan was noted for i*nvisibility, complexity, difficulty regarding estimation, dynamic nature, intangibility* and *regular upgrades*.
- *Bugs*. Many bugs were detected during the acceptance testing phase. All the bugs could have been eliminated on time in case the users were committed to conduct the acceptance test.

- *Difficulty to add people to a delayed project*. No deviation from the plan was noted.
- *Training*. No deviation from the revised plan was noted although the exercise was conducted at the contractor's site.

Environmental management

Social and political aspects were relevant for this project. The project involved the submission of an application for jobs using the Internet. Therefore, it relieved applicants of the need to go to a post office to send the application form. Time is saved by the applicant and the cost of postage is waived. Therefore, applicants (citizens) have time available for social activities.

The political dimension refers to the political benefits to be derived by developing citizen-centric applications. Such applications are being given high importance as they affect the lives of citizens by making services available online. They can benefit from online services round the clock and without the need to go to the offices that may be far from their residence. The availability of online services can also increase the personal computer and Internet penetration rate in the country.

As Mauritius is an African developing country, high power distance and moderate uncertainty avoidance is a matter of concern. These Hofstede's dimensions were taken into consideration during the management of the project. As such, the project manager had to intervene to counteract the effect of high power distance at the software development organisation as well as at the client's organisation. Similarly, during the execution of the project, risk was taken to adopt the latest technology. This was done with the result that the actual time for development exceeded the planned time. Therefore, as uncertainty due to change in technology was not avoided and the contractor made an attempt to use the latest technology, the project success was compromised to the extent that it was delivered late. Collectivism and masculinity/feminity dimensions were not found to be of importance in this project.

The PDCA cycle for Environmental management

In managing the environmental factors, the following inputs were transformed into outputs:

- List of environmental factors to be addressed.
- List of ways to deal with the factors.
- Historical data.

The transformation generated the following outputs:

- Satisfied team members.
- Safe software development environment.
- Lessons learnt.

The transformation of the inputs to the outputs was as per figure E.5.

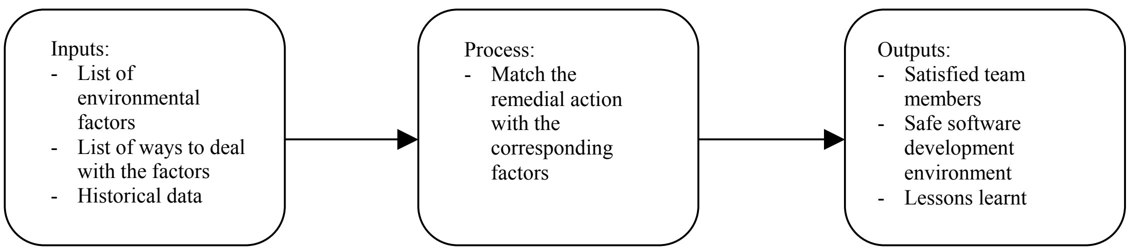


Figure E.5: Environmental management - Transformation of inputs into outputs

In the ESPM[3] for this project, the environmental factors were taken into consideration at all maturity levels. A list of factors was prepared and actions to be taken were defined. These factors were mapped onto four process groups, namely plan, do, check and act as shown in figure E.6.

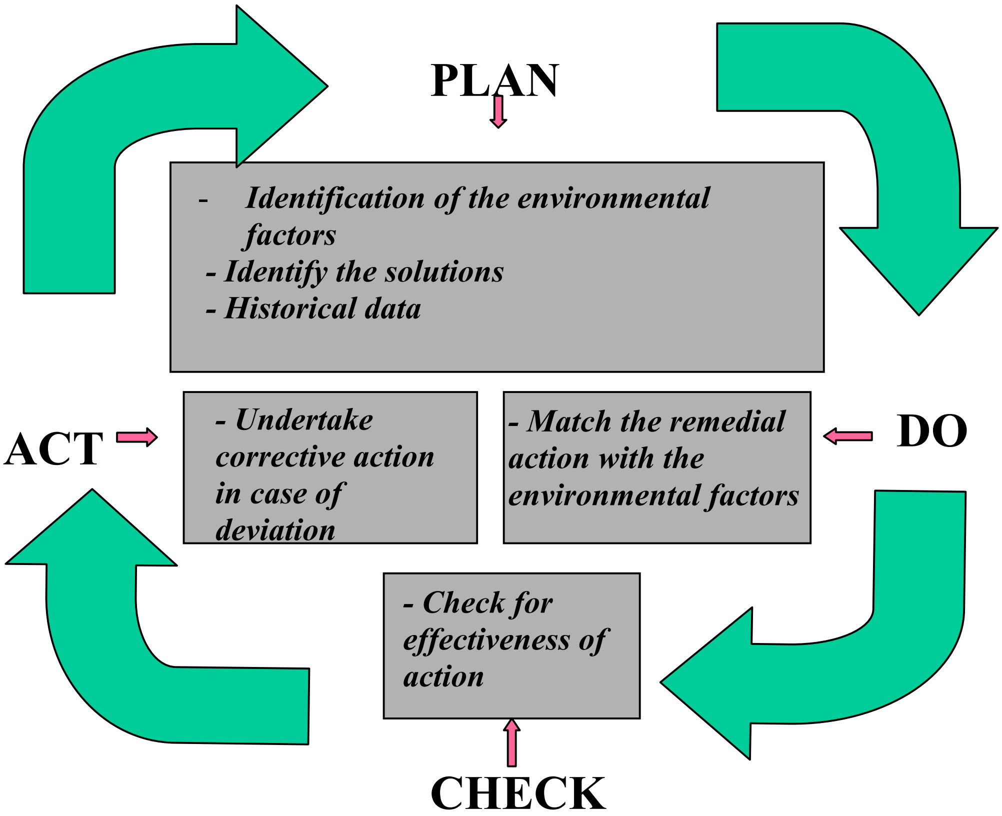


Figure E.6: PDCA cycle for Environmental management

Plan: Environmental management

The relevant factors were listed for consideration during the project execution. Social and political dimensions were identified for the project as it could affect the society by availing them of more free time and at the same time the political objective would be met by providing services online to the public (citizen-centric applications).

As more than 50 services were already available online (NCB, 2008), the proposed service for job application would be an added advantage to the citizens. Citizens were also already aware and exposed to this new channel of service delivery.

High power distance was noticed at both the software development organisation as well as at the client's organisation. Communication between top management and the software developers are very rare due to high power distance. This was also noticed between the top management and the users at the client's organisation. It was planned that the project manager would reduce the power distance between the parties concerned.

The project was expected to provide citizen-centric services like online job application, viewing of vacancies online and notification of the filling of vacancies. An attempt to counteract the moderate uncertainty avoidance was made by planning for the use of new technology while the project was underway.

Do: Environmental management

The plan for environmental management included the social and political dimensions. Social benefits were ensured by implementing a solution so that applicants could submit applications for jobs using the Internet round the clock and at the same time save on transport and postage costs as well as benefit from the availability of more time. At the same time, the efficiency of operations of that department would be enhanced through the software that would be implemented. The applications for jobs would be received in an electronic form ready for processing by the system. The applicants would benefit from reduced time in submission of their applications as well as from faster recruitment procedures after the closing date.

In addition, the political benefits of online delivery of services would be achieved. This new online service would add to the already available list of services provided by both private and public organisations.

The project manager had to intervene to reduce the high power distance at the client's organisation in order to secure management support and to facilitate the communication with the users. Intervention was also made within the software development organisation by addressing issues like leave and flexible working hours.

Uncertainty concerning the use of the new technology was not avoided as a new database management system was considered.

Check: Environmental management

Regularly, the social and political objectives planned had to be checked. Much attention was paid to the functionality of online delivery functionalities although other aspects had to be given priority for such environmental dimensions to be realised.

It was necessary to spend much time in conducting the acceptance testing phase in order to prevent bugs that would jeopardise the success of the project when it would be put into live operation. Any problem that would occur during a possible live run phase would discourage applicants to use the system in future and prevent political advantages from being derived.

As far as power distance was concerned, it was found that the project manager managed to keep all issues raised by the development team under control. These issues pertained mainly to leave and a flexibility in working hours. As the duration of the project was quite long, it was fair to provide such flexibilities in order to keep the staff motivated. The reduction of power distance at the client's organisation was also carried out as planned.

The use of the new database management system caused a delay in the development of the software. Although Mauritius is characterised by moderate uncertainty avoidance, it could be noted that an attempt was made to face the uncertainty.

Act: Environmental management

The detection of several bugs during the acceptance testing phase increased the users' concern about the software. Decision was taken by the contractor to extend the project duration in order to deliver a software product free from bugs although it was delivered late and the additional cost had to be absorbed.

Thus, more time was allocated to the acceptance testing phase so that this phase could be done well with the commitment of the user.

The high power distance prevailing in Mauritian organisations was an issue that had to be resolved by the project manager at both the users' side as well as at the software development organisation's side. This not only ensured the motivation within the development team but also contributed in establishing a conducive environment. The project manager had to be on his guard to be able to allow a trust relationship in order to know about the problems of the staff and to resolve them as and when they arose.

The fact that uncertainties were not avoided in this project contributed to a delay of the project. This delay could have been avoided by ensuring that training is provided in advance on new technologies before they are officially released. In the case of software, the exposure to beta versions need to be enforced. It is speculated that uncertainty avoidance is beneficial to Mauritian organisations.

Integration management

Integration testing was carried out by execution of the PDCA cycle so that the KPAs could be dealt with in a coordinated manner with the ultimate goal of achieving the overall objective of the project. In the ESPM^3, the integration management KPA ties all the other KPAs together so as to ensure that a common goal is attained.

The following inputs were considered in integration management:

- Other KPAs.
- Constraints.

The outputs generated were:

- Project results.

- Lessons learnt .

The transformation of the integration management inputs to the outputs is shown in figure E.7.

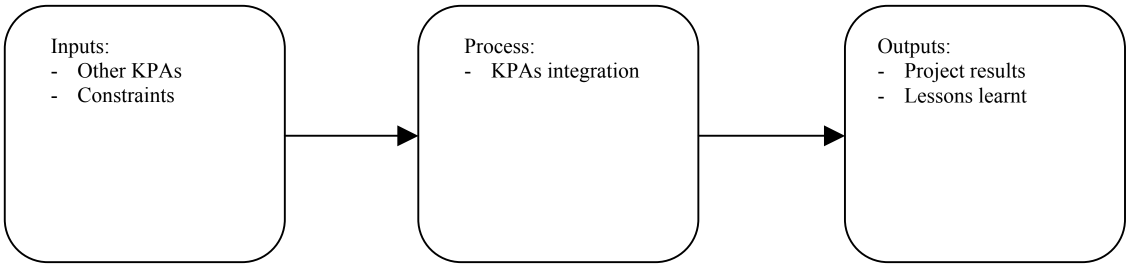


Figure E.7: Integration management - Transformation of inputs into outputs

The mapping of integration management onto the four process groups, namely plan, do, check and act was as shown in figure E.8.

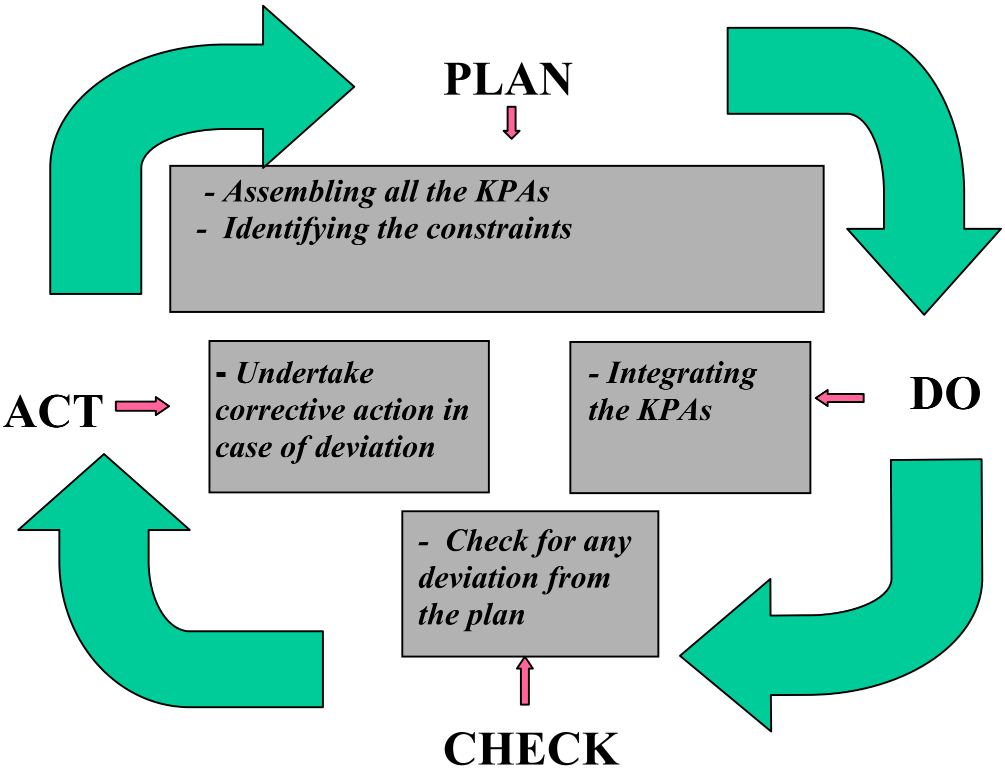


Figure E.8: PDCA cycle for Integration management

Plan: Integration management

The objective of the planning process group for integration management was to bind all KPAs together so that they could be dealt with in a coherent manner. In the first instance, the level 2 KPAs were found to be tightly bound. Any delay in the execution of the project caused the cost of development to increase as well. In order to maintain the quality of the product, additional effort was also required. The project manager had to keep track of all the KPAs (as per table 12.15) in order to facilitate the success of the project.

Table E.1: Integration management planning

<table>
<tr><td colspan="2">Continuous Process Improvement Group</td><td>Level 2 KPAs</td></tr>
<tr><td rowspan="3">Integration Management</td><td rowspan="3">Soft skills management
Change management
Software specific focus
Environmental management</td><td>Time management
Cost management
Quality management</td></tr>
<tr><td>Level 3 KPAs</td></tr>
<tr><td>Human resource management
Risk management
Contract management</td></tr>
</table>

The planning process group for each KPA was defined in parallel. All the plans were prepared to be executed simultaneously.

Do: Integration management

This process group is concerned with the execution of the plans for all the KPAs. All the plans from the previous process groups were executed in a coordinated manner. With the overall objective of a successful project delivery, each of the “Do” process group was executed, followed by the next process group.

Check: Integration management

The actual results were compared with the planned results in a coordinated manner.

Act: Integration management

After the comparison of the actual results against the planned results, any deviation was identified and action was taken. For instance, the loss of one employee during the acceptance testing phase resulted in a suitable staff member to act as a replacement and continue with the other activities of the project.

PUBLISHED PAPERS

TOWARDS A FRAMEWORK FOR EVALUATING STRENGTHS AND WEAKNESSES OF SOFTWARE PROJECTS

Sukhoo, A.[1], Barnard, A.[2], Eloff, M.M.[2] and Van der Poll, J.A.[2]

[1]Central Informatics Bureau
Royal Road, Paillotte, Quatre Bornes, Mauritius. Tel: +230 201 2011.
E-mail: aneeravsukhoo@yahoo.com
[2]UNISA
School of Computing, UNISA, South Africa. Tel: +27 12 429 6817.
E-mail: barnaa@unisa.ac.za, eloffmm@unisa.ac.za or vdpolja@unisa.ac.za

ABSTRACT

Currently software projects are considered by some as being no different from traditional engineering projects. Software project management methodologies are developed with a generic concept in mind as is the case with Prince2 and PMBOK. Although these methodologies have a wider scope, there are certain specificities tightly bound to software projects that warrant the need for, in particular, project management methodologies that focus on the development of software projects. This paper presents these specificities in terms of strengths and weaknesses of software projects in contrast to traditional engineering projects. These strengths are factors to be considered seriously and advantage should be taken of them in the management of software projects. We speculate that these strengths may indeed present a plausible framework for the future analysis of different software project management methodologies. At the same time, project managers ought to be cautious about the weaknesses inherent in software projects.

Keywords: Software, project management, strengths and weaknesses of software projects

INTRODUCTION

Many project management methodologies have been developed with a view to address all types of projects. These methodologies are often said to be generic in nature and are expected to cut across various disciplines (Cockburn, 2000). Software projects are defined in much the same way as traditional engineering projects. Under the term traditional engineering project, we understand those projects associated with the creation of engineering artefacts that are not of a software nature.

Some definitions of the term “project” to be found in the literature include:

- A human endeavour which creates change, is limited in time and scope, has mixed goals and objectives, involves a variety of resources and is unique (Andersen, Grude, Haug and Turner, 1987)
- A complex effort to achieve a specific objective within a schedule and budget target, which typically cuts across organisational lines, is unique and is usually not repetitive within the organisation (Cleland and King, 1983)
- A one time endeavour to do something that has not been done that way before (Smith, 1985)
- A temporary endeavour undertaken to create a unique product or service (PMI, 2000; Schwalbe, 2000).

Taking these definitions into account, and in particular the approach suggested by PMI (2000) and Schwalbe (2000), Marchewka (2003, p. 9) then claims that project management *“is the application of knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations from a project”*.

All these definitions are malleable to be applicable to software as well as traditional engineering projects. In spite of such common definitions, it is our contention that differences between software projects and traditional engineering projects readily show up. These definitions may furthermore impact on the way in which software projects are handled and managed as the influence of the human factor is often ignored. We furthermore speculate that the strengths inherent in software projects should be taken advantage of to steer such projects towards successful completion within the set budget, time and quality standards. However, software projects also have intrinsic weaknesses that may hamper the proper management thereof, if these are overlooked. Project managers must constantly be alert to overcome or mitigate these weaknesses.

The remainder of this paper is structured as follows: Section two discusses typical strengths of software projects as opposed to traditional engineering projects. The third section elaborates on some of the weaknesses normally associated with software projects. We conclude this paper in section 4 by noting that the specificities associated with software projects, should be taken advantage of to steer such projects to successful completion.

STRENGTHS OF SOFTWARE PROJECTS

Over the past two decades, several problems associated with software project development, contributed to the general failure of these projects to live up to user expectations, they were commonly delivered late, and they mostly ran over the set budget. The Standish Group studied 13,522 projects and reported thereon in the EXTREME CHAOS (2000) report. In particular this study determined that 23 percent of the surveyed projects failed, 49 percent did not meet the requirements and only 28 percent succeeded. Marchewka (2003, p. 6) identified the following as some of the contributing factors to software project failure: *incomplete requirements*, *lack of user involvement*, *lack of resources*, *unrealistic* user *expectations*, *lack of executive support*, and the *lack of* proper *planning*, among others.

Despite the problems related to managing software projects as explicated above, we advocate that software projects have unique strengths when compared, for example, to traditional engineering projects. Project management methodologies should emphasise these strengths to address some of the possible problems associated with software projects and their management. The generic nature of project management methodologies is to a large extent suitable for most projects, however in many cases customisation may be required in software project development.

An insight into the strengths associated with software projects, is important to focus attention on the relevant areas. In this section we investigate some of the more important strengths (Brooks, 1995; Hughes and Cotterell, 2002) associated with software project development in general:

FLEXIBILITY

The inherent characteristics of software artefacts are often nebulous, and this more often than not imply that software can be modified with greater ease when compared to existing traditional engineering artefacts (Hughes and Cotterell, 2002). Customisation of software is frequently performed in an attempt to shorten the development schedule.

Whenever a software module is developed and presented to a user, certain modifications are normally requested. These requests may range from minor to major modifications, especially if the user requirements were not captured properly during the analysis phase. It is rare occurrence that a software product has to be redesigned and developed in its entirety due to erroneous (in case of user resistance), or badly captured, user requirements. Furthermore, it is often the case that bugs or other types of unanticipated errors are identified only when the software has been in operation for some time already, and consequently has to be fixed. This is normally carried out with an acceptable degree of ease and flexibility to the commensurate level of correctness.

In contrast to the general flexibility of software products discussed above, traditional engineering artefacts do not lend themselves easily to modifications without structural impact. A bridge, building, machinery or an electronic piece of equipment may not be easily modified, if this is even at all possible. A redesign of the products may, therefore, be cost-prohibitive and hence unfeasible to undertake. This reflects an advantage that the flexibility associated with software product alteration has over that of traditional engineering products.

MOBILITY

Software can be transported at low or no cost at all since it can be rapidly transferred from one locale to another by utilising existing communication options. Trial versions of software may readily be downloaded over the Internet across thousands of kilometres at normal Internet access cost. If the user is satisfied with the trial version of the software, (s)he may decide to procure the registered version. Software can also be transferred using variable sized diskettes, CD ROMS, etc.

Traditional engineering artefacts on the other hand, very often cannot be easily relocated from one physical location to another. This is clearly the case for bridges, dams and concrete buildings. Once built, they stay in place, at least for the foreseeable future (assuming normal circumstances).

REPLICATION

Replication of the same software product is very common and this feature accounts for reduction in cost and duration of implementation. Examples include the replication of freeware like UNIX and Linux together with commercially available registered software such as Windows XP®, Office XP® and StarOffice® under license agreements. Replication of software products is usually associated with minimum possible effort and can quickly progress round the globe. This is but one of the reasons that explains the prosperity of major software developers such as Microsoft™ and Oracle®. Project managers should bear in mind that the low cost of software replication could offer a competitive advantage.

Traditional engineering projects, however, generally yield products that cannot be replicated as readily as is the case with software projects (accept for some mass produced commercial items). Development of a similar traditional engineering product may require as much effort, time and cost as the original one.

SCALABILITY

Software is scalable and can be adapted to new and emerging needs. A transaction processing system may be developed to record the day-to-day activities of an organisation. This system may then be used to develop a Management Information System for middle management to produce summarised structured reports. In order to analyse unanticipated situations, middle management may request a Decision Support System, which draws on the data from the underlying database. Higher up in a hierarchy, top management is especially concerned with an Executive Information System capable of presenting information in an executively summarised form (O'Leary and O'Leary, 2000).

Scalability may not be the case for the majority of traditional engineering products. Another floor cannot be readily added to a building unless it has been planned beforehand, in which case supporting iron bars and columns of the necessary material of predetermined diameter must be carefully selected.

BACKUP

Backup is an effective means to minimise data and software loss. The backup procedure in the software domain is carried out at minimum cost and effort while the backup of a traditional engineering product may prove to be too expensive to maintain. Software backup is, among others, performed for security purposes and to maintain the integrity of data (Pfleeger and Pfleeger, 2003). In the case of software and associated data loss resulting from hardware malfunction, deliberate corruption or natural disasters, recovery procedures can be initiated from the backup version. Clearly, this is not the case for traditional engineering artefacts.

REUSABILITY OF COMPONENTS

Program modules used in one project can be reused a number of times in other projects, thusly reducing development efforts resulting in time and cost savings. Reusability of components is a common practice in software development. Software developers ought to be well aware of the importance of a library of routinely used modules. It should not be necessary to "reinvent the wheel" whenever the same functionality must be incorporated in other software.

In the case of a traditional engineering product, however, a specific part cannot be removed from the said product and used in another product without usually rendering the original product non-operational.

POSSIBILITY OF USING PROTOTYPES

Prototyping can be used to construct a final software product. This initiative reduces the effort and eventually the cost associated with the development of a software product. There are various prototype options to be used in software project development. An example includes evolutionary prototypes that are initially presented to the user during early development stages to clarify any misunderstanding between the user and the development teams. The envisioned result of this activity is to enable the capture of highly focused user requirements. This prototype is a simplified version of the final product that are rapidly developed and which will act as a model to show the user what can be expected at the final stage(s) of the project. It is in fact a scaled down version of the final product. An evolutionary prototype is modified in a stage-by-stage manner until it is approved by the user for final implementation (Davis, 1992). Development effort is not necessarily wasted since the evolutionary prototype is not discarded after it has been presented to the user. This may obviously not be the case for throwaway prototypes.

A traditional engineering prototype on the other hand is a representation that usually cannot easily be evolved to constitute the final product. For example, a model of a ship, a building or a car cannot readily be used to build the end product. Users will normally be charged for such development efforts, even if these models are discarded.

SOFTWARE EVOLUTION

A software project can evolve over time, resulting in new releases that outperform older versions. New releases tend to take advantage of prevailing technology where software evolution is then commonly used for improving software. For example, the Microsoft Windows® operating system and Microsoft Office® products are subject constantly to new releases which incorporate new functionalities, better user interface designs and performance and also take advantage of new technologies like new and faster hardware as well as better communication facilities etc. An example of this is provided by Microsoft's™ Next-Generation Secure Computing Base (NGSCB), code-named *"Longhorn"*®, a new hardware and software design that enables new kinds of secure computing capabilities for providing *"enhanced data protection, privacy and system integrity"* (Microsoft Next-Generation Secure Computing Base – Technical FAQ, 2003). In particular, Microsoft™ describes its Security Support Component (SCC) as *"a new PC hardware component that will be introduced as part of the NGSCB architecture … is a module that can perform certain cryptographic information and securely store cryptographic keys … that provide seal storage and attestation function"* (Microsoft Next-Generation Secure Computing Base – Technical FAQ, 2003). Microsoft™ furthermore claims that the addition of such hardware and software technology to the Microsoft Windows® platform has the associated advantages of offering increased security, privacy and system integrity features.

Traditional engineering artefacts, like bridges and buildings are not necessarily built with an attempt to evolve them or to take advantage of new technologies to improve them. Where the products are evolved to newer models they do not readily replace the original ones. Usually the older and newer versions coexist with each other. Examples include new models of cars and more recent versions of electronic equipment. However, in contrast to software products, traditional engineering products are generally used over a longer period.

WEAKNESSES OF SOFTWARE PROJECTS

Software projects suffer from certain weaknesses when compared to traditional engineering projects. These weaknesses must be carefully dealt with in order to achieve successful software project completion. Therefore, any proposed software project management methodology should ideally take cognisance of inherent weaknesses in software projects (Olson, 2004).

Examples of weaknesses related to the development of software projects, are:

INVISIBILITY

In the case of the construction of a physical artefact such as a bridge, the progress of work can actually be seen. In the case of software projects this progress is not immediately visible (Hughes and Cotterell, 2002, p.3). One major weakness is the invisibility or poor visibility of progress during the development of a software project. The user may not be able to conceptualise the software until it is complete. If the user is not satisfied, a great deal of effort may have already been expended, ultimately resulting in project failure. Strategies of software project management must therefore strive to make the invisible visible through the judicious use of appropriate software or project management tools and methods.

In contrast to the invisibility usually associated with the development of software products, traditional engineering product development on the other hand is normally visible as it lends itself to observation of tasks while they are carried out.

Complexity

Software artefacts are inherently more complex than traditional engineering artefacts (Brooks, 1987). Hughes and Cotterell (2002, p. 3) states that per dollar, the complexity associated with software project development is more than that of a traditional engineering project. At the inception of a project, inaccurate or ambiguous information gathered from the users may constitute a major threat to subsequent stages of the software development process. A logical error may propagate throughout the development process and can go undetected until the last stages. In 1976 already Fagan (1976) stated that errors due to the complexity inherent in software projects, can even go undetected for a long time after the final product has become operational, and that the impact may then be considerable.

In traditional engineering projects, complexity is relatively lower owing to better visibility of progress and usually well-understood requirements from the onset.

DIFFICULTY REGARDING ESTIMATION

Estimation of the software production process in terms of cost and duration is perceived as a difficult task (Hughes, 1996). An inaccurate estimation may lead to cost overrun, failure to meet deadline or degradation in quality (Yamaura and Kikuno, 1999). Several methods for software estimation include Expert Judgement, Estimation by Analogy, COCOMO, Function Point Analysis, etc. (Schach, 2002). Agarwal, Kumar, Mallick, Bharadwaj and Anantwar (2001) points out that the causes of *"poor and inaccurate estimation are: (a) imprecise and drifting requirements; (b) new software projects are nearly always different form the last; (c) software practitioners don't collect enough information about past projects; and (d) estimates are forced to match the resources available"*. However, during the early stages of the project, it is often difficult to estimate the number of components constituting the product and the time to be spent on each. A start-up software development company will have to test an estimation method for some time before making appropriate estimations. It is often the case that users are not satisfied with high software costs proposed by software developers. This is probably due to invisibility of the amount of work required.

Traditional engineering artefact development has been in operation for eons and consequently the visibility of the amount of work concerned, price of components, and existing well-defined complexity measures, may account for relatively easy negotiations regarding cost and time schedules between the user and the developer.

DYNAMIC NATURE

The dynamic nature of software becomes a limitation in certain cases where new versions are no longer compatible or suitable to run on older versions of operating systems software or hardware. For instance, a 16-bit software product may not function on a 32-bit operating system or Microsoft Office XP® will not run on a personal computer having an 80486 processor. Software developers are also reluctant to support older versions of software and therefore, users are forced to review hardware and software regularly and upgrade both at the same time.

INTANGIBILITY

The intangibility feature associated with software makes it difficult for potential clients to purchase a certain product (Olson, 2004). Sometimes free trial versions are offered to assist clients to reach a decision regarding purchase of the registered version. When this is not the case, especially when the requirements of the user cannot be matched with existing software, the software has to be tailor-made. If an evolutionary prototyping method is not used, the user will not be able to appreciate the eventual functionality of the software until it is completed.

On the other hand, the tangibility associated with a traditional engineering product, is evident throughout the construction process. The client may watch the progress of the project on site and even communicate any issues of dissatisfaction to the developer immediately. This may enhance client satisfaction at the closure of the project.

REGULAR UPGRADES

Regular upgrades improve the quality, performance and efficiency of software, but in the long run becomes expensive. Sometimes older versions of software are no longer maintained or the maintenance service is offered at a higher cost. The user is forced to bear this cost to stay competitive. Often an upgrade regarding software as well as hardware is required yearly, however, this may not always be affordable due to the high cost involved and the frequency of said upgrade. For example, Microsoft Office® is upgraded virtually annually and organisations with a large number of personal computers may find it too costly to upgrade the software every year in exchange for increased productivity.

Traditional engineering products do not require frequent upgrades, as is the case with software products. Once a building is completed, it is planned to be in that state for years. One can argue with a degree of certainty that additional floors in the building will not be added annually.

BUGS

It is often observed that software suffers from bugs and flaws from the time it is released, requiring the software developer to be called upon to correct them. Sometimes, this affects the confidence of a user towards a software developer. One can speculate that the greater the number of bugs, the greater the loss of confidence. Therefore, a software developer should aim at eliminating these bugs as soon as possible requiring additional effort throughout the development and testing phases. Bugs can often be attributed to the inherent complexity of software, as argued in a previous subsection on complexity.

On the other hand, flaws in traditional engineering products are not so common after release due to a number of factors. These factors include, but are not limited to, the better visibility during the development process, an acceptable level of complexity and existing formal methods to assist the development process that are entrenched in the subject theoretic literature, for example refer to the Whall verification debate (Le Charlier and Flener, 1998; Woodcock and Davies, 1996).

DIFFICULTY TO ADD PEOPLE TO A DELAYED PROJECT

It is difficult to meet project deadlines by adding new people towards the end of the development cycle of a software project. According to Brooks' Law, *"putting more people on a late job makes it later"* (Brooks, 1995). Generally new people added at late stage during software development, has to spend time learning what has been completed up to that point. Additional effort is also required regarding management, co-ordination and communication.

It is often easier to add new people on a traditional engineering project that is behind schedule. For example in the construction of a bridge or a building, adding more people will generally speed up the construction process. However, the number of additional people must be carefully chosen so as not to impact on management overheads.

Training

Software projects, most often if not always, need training sessions to be incorporated to use the end product (software). In the case of traditional engineering products, this may often not be the case. Generally, the user does not require extensive training to use a bridge, building or road. Sometimes, following the instructions presented in a user manual, will be sufficient to operate a piece of equipment.

CONCLUSION AND FUTURE WORK

Project management is not a new concept! Throughout the documented history of the human race there have been projects that attempted to do more than a single person could accomplish alone. Think of examples such as the pyramids in Egypt, the cathedrals in Europe, and the Great Wall of China! Nowadays effective management of projects in the Information Technology environment is seen to be increasingly important as a discipline.

However, project management methodologies, which are generic in nature, have been developed to help organisations cope with projects across various disciplines. Although these methodologies have a wider scope, there are certain specificities tightly bound to software projects. These specificities highlight the need for project management methodologies that focus on the development of software projects. In this paper we investigated and documented the strengths and weaknesses associated with software projects as opposed to traditional engineering projects.

This paper constitutes part of a larger research project focusing on a critical analysis of the situation pertaining to software project management in Mauritius. The use of project management methodologies, tools and techniques faced by software developers in Mauritius in particular, were analysed and reported on in another paper (Sukhoo, Barnard, Eloff and Van der Poll, 2004). Given the assumption of economic rationality and cultural differences and the need to cope with political and community demands on a project's resources (Murithi and Crawford, 2003), a new software project management methodology for Mauritius, is envisaged. With regards to the strengths and weaknesses of software projects explored in this paper, such methodology should take cognisance thereof to exploit the strengths and eliminate the weaknesses where possible.

REFERENCES

1. Agarwal, R.; Kumar, M.; Mallick, S.; Bharadwaj, R.M. and Anantwar, D. (2001). Estimating software projects. *Software Engineering Notes*; 26(4), 60-67.
2. Andersen, E.S.; Grude, K.V.; Haug, T. and Turner, J.R. (1987). *Goal directed project management*. Kogan Page/Coopers and Lybrand, London, UK. Cited in Turner, J.R. and Müller, R. (2003). On the nature of the project as a temporary organisation. *International Journal of Project Management;* 21(1), 1-8.
3. Brooks, F.P. (1995). *The mythical Man-month: Essays on software engineering,* anniversary edition; Addison-Wesley: Reading, MA, US.
4. Brooks, F.P. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Computer;* 20(4), 10-19.

5. Cleland, D.I. and King W.R. (1983). *Systems analysis and project management*. Mc Graw-Hill: New York, US. Cited in Turner, J.R. and Müller, R. (2003). On the nature of the project as a temporary organisation. *International Journal of Project Management;* 21(1) 1-8.
6. Cockburn, A. (2000). Selecting a project's methodology. *IEEE Software*; 17(4), 64-71.
7. Davis, A. (1992). Operational prototyping: A new development approach. *IEEE Software;* 9(5), 70-78.
8. Extreme Chaos Report. (2000). *Extreme Chaos Report*. Compiled by The Standish Group. Accessed 20.11.2003, online at: http://www.standishgroup.com/.
9. Fagan, M.E. (1976). Design and code inspections to reduce errors in program development. *IBM Systems Journal*; 15, 182-211.
10. Hughes, R.T. (1996). Expert Judgement as an Estimating Method. *Information and Software Technology*. 38, 67-75.
11. Hughes, B. and Cotterell, M. (2002). *Software Project Management,* 3rd edition. McGraw-Hill: London, UK.
12. Le Charlier, B. and Flener, P. (1998). Specifications are necessarily informal or: Some more myths of formal methods. *The Journal of Systems and Software*; 40(3), 275-296.
13. Marchewka, J.T. (2003). *Information technology project management: Providing measurable organizational value*. John Wiley and Sons: Hoboken, NJ, US.
14. Microsoft Next-Generation Secure Computing Base – Technical FAQ (2003). Accessed 20.11.2003, online at: http://www.microsoft.com/technet/security/news/NGSCB.asp?frame=true.
15. Murithi, N. and Crawford, L. (2003). Approaches to project management in Africa: implications for international development projects. *International Journal of Project Management*; 21(5); 309-319.
16. O'Leary, T.J. and O'Leary, L.I. (2000). *Computing Essentials*. McGraw-Hill: New York, US
17. Olson, D.L. (2004). *Introduction to information systems project management*, 2nd edition. McGraw-Hill/Irwin series: Boston, MA, US.
18. PMI 2000. *A guide to the project management body of knowledge*, 4th edition. Project Management Institute: Newton Square, PA, US. Cited in Turner, R.J. and Müller, R. (2003). On the nature of the project as a temporary organisation. *International Journal of Project Management;* 21(1) 1-8.
19. Pfleeger, C.P. and Pfleeger, S.L. (2003). *Security in Computing*. Prentice Hall: Upper Saddle River, NJ, US.
20. Schach, S.R. (2002). *Object-oriented and classical software engineering*. 5th edition, McGraw-Hill: New York, US.
21. Schwalbe, K. (2000). *Information technology project management.* Course Technology, Thomson Learning: Cambridge, MA, US.
22. Smith, B. (1985). Chapter in book: Project concepts, in *Effective project administration*. Institution of Mechanical Engineers: London, UK. Cited in Turner J.R. and Müller, R. (2003). On the nature of the project as a temporary organisation. *International Journal of Project Management;* 21(1) 1-8.
23. Sukhoo, A.; Barnard, A.; Eloff, M.M. and Van der Poll, J.A. (2004). A survey of project management tools, techniques and methodologies used in mauritius: the current status. To appear in the *Proceedings of the 2004 PMSA Global Knowledge Conference: "Current Research" Stream.*
24. Woodcock, J. and Davies, J. (1996). *Using Z: Specifications, refinement, and proof.* Prentice-Hall: London, UK.
25. Yamaura, T. and Kikuno, T. (1999). A framework for top-down cost estimation of software development. *Proceedings of the twenty-third annual international Computer Software and Applications conference*, 322-323.

A SURVEY OF PROJECT MANAGEMENT TOOLS, TECHNIQUES AND METHODOLOGIES USED IN MAURITIUS: THE CURRENT STATUS

Sukhoo, A.[1], **Barnard, A.**[2], **Eloff, M.M.**[2] **and Van der Poll, J.A.**[2]

[1]Central Informatics Bureau
Royal Road, Paillotte, Quatre Bornes, Mauritius. Tel: (230) 201 2011.
E-mail: aneeravsukhoo@yahoo.com
[2]UNISA
School of Computing, UNISA, South Africa. Tel: (27) 012 429 6817.
E-mail: barnaa@unisa.ac.za , eloffmm@unisa.ac.za or vdpolja@unisa.ac.za

ABSTRACT

Developing countries are very often faced with a shortage of skilled staff, difficult economic and social conditions, weak political institutions, deeply rooted cultural and religious beliefs that all have an impact on development in various disciplines (Nessan, 1993). This paper critically analyses the situation of software project management in Mauritius with a view for further research in the field. Data sets for statistical analysis were collected using two questionnaires, one aimed at gathering information from software development companies and/or staff and another one targeting users of computer software. The use of project management methodologies, tools and techniques, considered as important for assessment by Ralph and Ludin (1996), faced by software developers in Mauritius in particular, were analysed. User satisfaction was one of the criteria analysed by the client questionnaire.

Based on the analysis of the surveys conducted, it was found that there is room for improvement regarding software project management in Mauritius and we speculate that this will also be the case for other regional developing countries. Methodologies developed for use in western countries may not be totally suitable for developing countries. Furthermore, given the assumption of economic rationality and cultural differences and the need to cope with political and community demands on the project's resources, a new project management methodology to be developed to ensure that developing countries, in particular Mauritius, are not left apart, is indicated (Murithi and Crawford, 2003).

Keywords: Software; Project management; Software project management methodologies, tools and techniques; Surveys; Developing countries

INTRODUCTION

Economic rationality is often assumed and cultural differences, political situation and social conditions are usually ignored when dealing with project management. Hofstede (1983) points out that *"project management can be related to differences in national cultures"* and Adler (1983) raises concerns regarding cross-cultural management. Murithi and Crawford (2003) found that *"in Africa, theorists and practitioners in many disciplines are increasingly wary of adopting concepts of (a) western origin"*. Even Turner (1993) as cited by Murithi and Crawford (2003) observed that *"contrary to the common belief that western-oriented techniques of project management are just straight forward procedures that anyone can learn and implement, there are considerable cross-cultural problems in using the approach in non-western countries"*.

The purpose of this survey is to analyse the situation of software project management in developing countries, and Mauritius in particular. A statistical approach is followed regarding data gathered from software development companies and/or staff and users of such software. Mauritius, as is the case in other developing countries are either using methodologies of western origin, or are not using any methodology at all. The information gathered from the questionnaires, after analysis, reveals that project managers in Mauritius are not at present entirely comfortable with the methodologies in use, given that the latter does not appear to yield expected success rates.

To achieve this goal, we briefly consider the research methods adopted in this study in section 2 of this paper while we focus on the analysis of the survey results in section 3. We conclude the paper by observing in section 4 that project management concepts, especially of western origins, may not be universally applicable.

RESEARCH METHODOLOGY

Two questionnaires were designed for the data collection activity. The survey was carried out between July 2003 and November 2003. The target group for one questionnaire focused on software development companies and/or staff. The questionnaire aims at gathering information about the respondents' awareness regarding prevailing project management methodologies in the Information Technology arena. It furthermore surveyed tools and techniques in use as well as the level of success of projects within their organisations with respect to the project parameters such as time, cost and quality. Out of a total of 80 questionnaires sent to software companies in Mauritius and other regional developing countries, 16 completed questionnaires were received. 10 completed survey forms were received from Mauritian companies while 6 forms came from other regional developing countries like South Africa, India, Kenya and Zimbabwe. As the majority of responses were received from Mauritius, we focus in this paper on the status quo of the use of software project management methodologies in Mauritius. It is, therefore, important to note that 10 responses were received from the 21 questionnaires sent to software development companies in Mauritius in particular. Many project managers in Mauritius were contacted by phone to fill in the questionnaires, and it was particularly noteworthy that top managers were reluctant to provide substantial information. Subsequently in section 3 below we present an analysis of the data gathered from this survey with reference to the Mauritian environment. We intend to conduct the survey in the other developing regional countries, with more rigour in the future.

Another questionnaire was dispatched to the users of software that were developed in Mauritius. The objective of this questionnaire was to investigate user satisfaction with respect to software developed and supplied by software companies in Mauritius. A total of 60 users and/or clients were contacted and all of them responded positively to the request for information. Another objective of the questionnaire was to confirm whether or not software companies were making the necessary efforts to ensure user satisfaction with respect to timely delivery, optimum cost and an acceptable quality standard for the product concerned.

ANALYSIS OF SURVEY RESULTS

The data from these questionnaires were used to determine:

- the awareness of existing project management methodologies in use by software developers and/or project managers;
- tasks addressed by the methodologies in use;
- project management tools in use;
- software development approach(es) in use;
- quality management standards in use;
- information reports received from companies regarding projects suffering from failure to meet deadline, budget overrun and user dissatisfaction;
- reasons supplied by the respondents regarding their perception of the deviation in project achievement;
- quality of software from the users' point of view;
- causes of failure of projects to meet expected objectives according to users; and
- factors to improve user satisfaction supplied by the respondents.

ANALYSIS OF SOFTWARE DEVELOPERS AND/OR COMPANIES SURVEY RESULTS

The data collected was analysed and is represented graphically in this paper by considering the percentages as well as mean scores. For some questions, a number from 1 to 5 was requested (1 – strongly agree; 2 – Agree; 3 – Neutral; 4 – Disagree; 5 – Strongly Disagree). The mean score was the average value calculated for a particular question received from all Mauritian respondents. This result shows the tendency of the response for a particular question. In the following subsections we focus only on the data evaluation of the 10 respondents received out of a possible 21 questionnaires distributed in Mauritius in particular.

Awareness of the Existence of Project Management Methodologies

The chart in figure 1 shows that the majority of the methodologies have scored less than 50% regarding their awareness among software developers. This chart indicates that the most popular methodology used in Mauritius is Prince 2 (OGC, 2002; CCTA, 2002), while Euromethod (Helmerich, 2002; Hughes and Cotterrel, 2002) and BS6079 (Hughes and Cotterrel, 2002) are the least known among software development personnel in Mauritius. It is, therefore, reasonable to assume that a significant number of software development personnel in Mauritius are familiar with methodologies such as Euromethod, ISO 12207 (Hughes and Cotterrel, 2001; Moore, 2002), PMBOK (Hughes and Cotterrel, 2001) and BS6079.

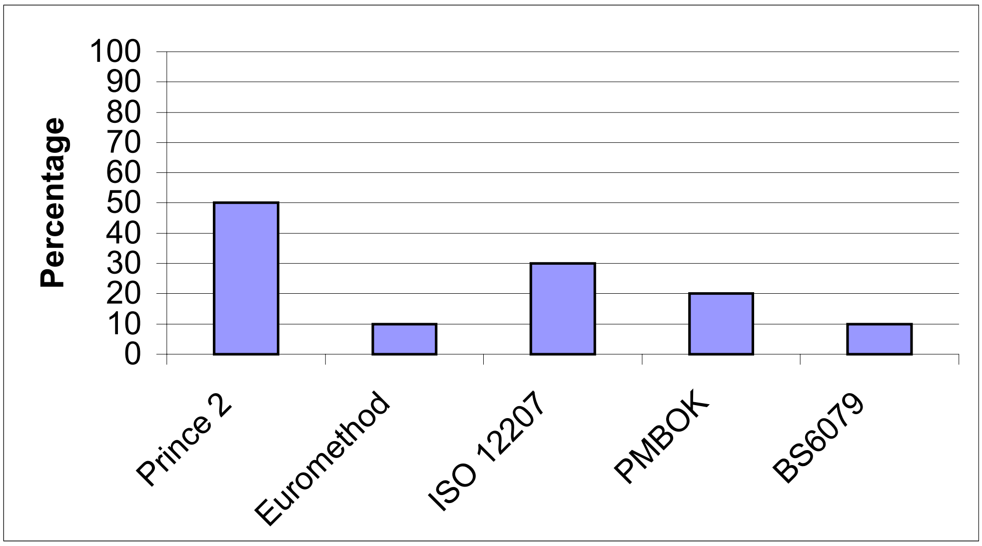


Figure 1. PM Methodologies awareness.

When software developers were polled regarding the methodology they were using, a significant number, 60%, of the respondents mentioned that they were not using any methodology for their software development process. However, it is our perception that some internal guidelines exist so that new entrants to the organisation can at least follow the relevant procedures, note that this perception was not explicitly measured by our survey.

We speculate that some reasons for the absence of a formal methodology include, among others:

- limited time allocated for completion of a project;
- software professionals are not provided with appropriate training regarding software project management;
- scarce staff resources.

All respondents, where a project management methodology is in use, claim to support multi-project management and the average number of projects handled concurrently amounts to 3.6. This appears to be a reasonable average, especially where large projects are concerned.

Tasks Addressed by Methodology in Use in The Organisation

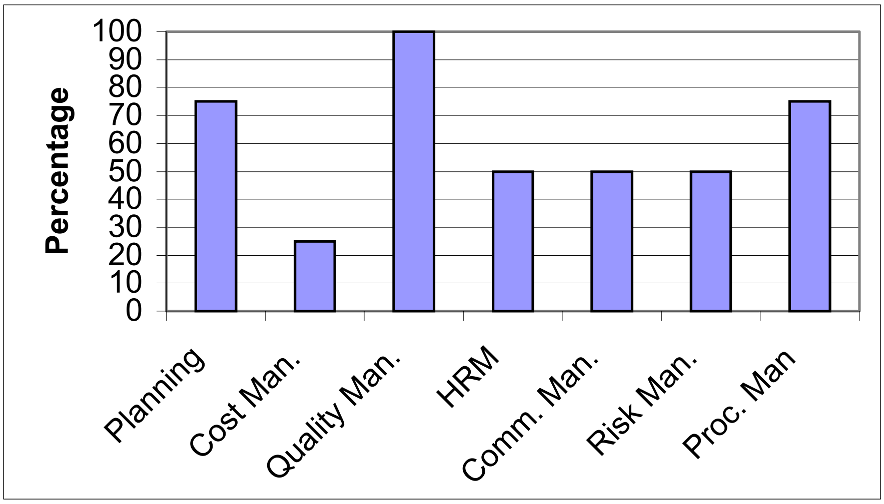
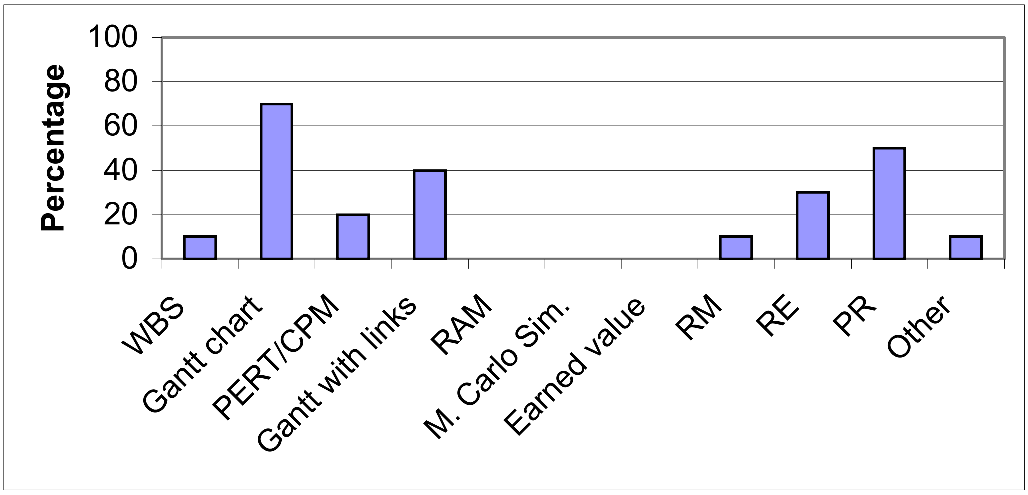


Figure 2. Tasks addressed by Methodology.

From the survey data it seems that the use of a project management methodology emphasises quality management (percentage = 100%), while cost management scores 25%. From the gathered data, planning and procurement management are also well represented at around 75%. Human resource management, communications management and risk management scored 50%. It is, however, possible that human resource management, communications management and risk management are addressed separately rather than by the methodology employed, although grouping all these tasks under the same umbrella would be helpful.

Project Management Tools Used

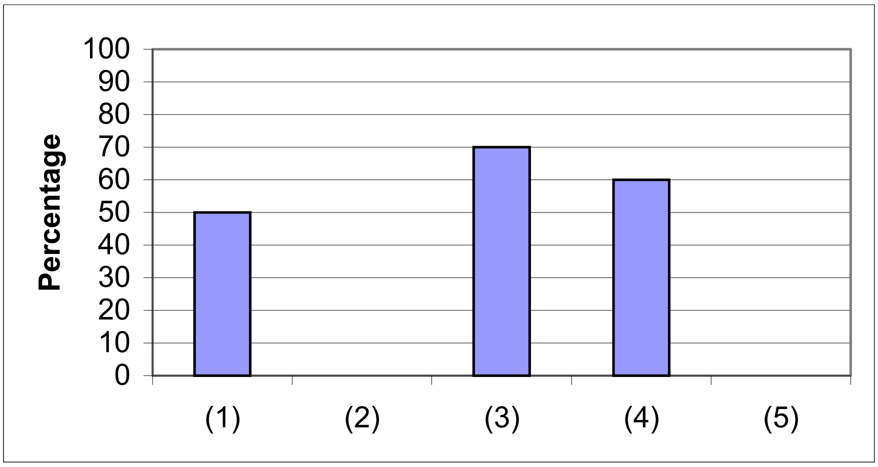


Legend:	WBS - Work Breakdown Structure
PERT/CPM	Project Evaluation and Review Technique/Critical Path Method
RAM	Risk Assessment Matrix
M. Carlo Sim.	Monte Carlo simulation
RM	Responsibility Matrix
RE	Resource Estimate
PR	Progress Report

Figure 3. PM Tools used.

The data gathered indicates that Gantt charts (around 70%), resource estimates (around 60%) and progress reports (around 50%) are popular tools in the management of software development in Mauritius. Monte Carlo simulation seems not to be frequently used by software developers. It appears that in Mauritius, software developers choose Software Project Management tools that do not take much time to implement in practice. This is possibly due to shortages of staff and the difficulty to allocate sufficient time for the use of software project management tools.

Software Development Approach

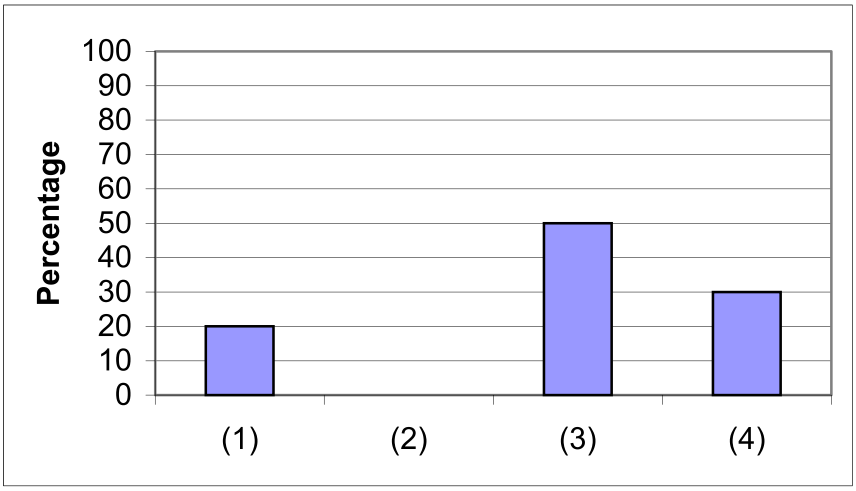


Legend:
(1) - Classical System Development Life Cycle (e.g. based on the Waterfall model)
(2) - System Development Life Cycle (SDLC)
(3) - 4GL (4th Generation Language) techniques
(4) - Prototyping
(5) –Other (e.g. formal methods)

Figure 4. Software development approach in use.

From the data it is found that most software companies prefer relatively short development cycles, where 4GL techniques are best preferred, 70%, followed by prototyping, 60%. The Classical System Development Life Cycle is used less, about 50%. Software with shorter development cycles, such as prototyping, have the added advantage that users can be given a “sense” of the final product. It was noted that companies in Mauritius did reportedly not use SDLC. Software developers and/or companies are generally concerned with the development of software within the minimum period of time.

Quality Management Standards

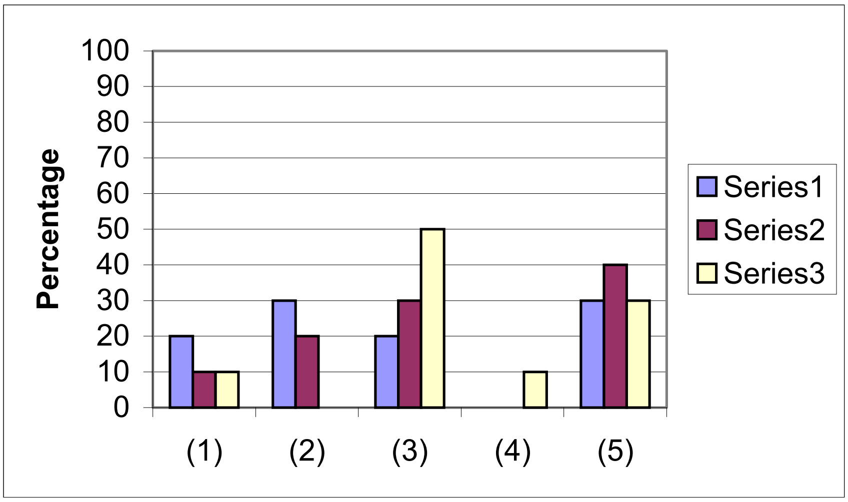


Legend:
(1) - International Standards Organisation (ISO)
(2) - Total Quality Management (TQM)
(3) - A standard developed in-house
(4) - No standard is used

Figure 5. Standards used.

The chart in figure 5, based on respondent information, reveals that 50% of software companies in Mauritius prefer to develop and use their own standards. This is possibly due to a lack of expertise with regards to international standards and difficulty to allocate sufficient time to investigate these standards given the scarcity of qualified staff. However, to extend their trade to other countries, an international standardised approach would certainly present Mauritian companies with a competitive advantage. Software professionals must take note that software project management has also been used for ISO 9000 certification (Ralph, 1999).

Projects Suffering from Failure to Meet Deadlines, Budget Overrun and User Satisfaction



Series 1 represents projects suffering from failure to meet the deadline.
Series 2 represents projects suffering from a budget overrun.
Series 3 represents projects suffering from failure to satisfy user requirements

Figure 6. Projects affected by deadlines, budget and failure to meet user satisfaction.

The approximate annual percentage of projects that fail to meet deadlines, suffer from budget overrun, or fail to satisfy user requirements, were categorised as follows:

- above 75%
- between 50% and 75% (both inclusive)
- above 0% but less that 50%
- 0%
- no response

Figure 6 indicates that:

- 20% of companies sampled claimed that more than 75% of their projects annually failed to meet the set deadlines
- 30% of companies sampled claimed that between 50% and 75% of their projects annually failed to meet the set deadline
- 20% of companies sampled claimed that less than 50% of their projects annually failed to meet the set deadline
- none of the sampled companies claimed that none of their projects annually failed to meet the set deadline
- 30% of companies sampled did not respond to this question.

These figures are quite alarming and we contend that this implies that improvement must be brought about in this area. We, therefore, suggest that better planning needs to be effected and that closer monitoring has to be carried out. It also appears that some companies are reluctant to disseminate the above kind of information.

Regarding budget overrun:

- 10% of companies sampled claimed that more than 75% of their projects annually suffered from budget overrun
- 20% of companies sampled claimed that between 50% and 75% of their projects annually suffered from budget overrun
- 30% of companies sampled claimed that less than 50% of their projects annually suffered from budget overrun
- none of the sampled companies claimed that none of their projects annually suffered from budget overrun
- 40% of companies sampled did not respond to the question.

Again, we deduce that much improvement is required in this area.

Regarding projects suffering from failure to satisfy user requirements, we note the following:

- 10% of companies sampled claimed that more than 75% of their projects failed to satisfy user expectations
- none of the sampled companies claimed that between 50% and 75% of their projects failed to satisfy user expectations
- 50% of companies sampled claimed that less than 50% of their projects failed to satisfy user expectations
- 10% of companies sampled claimed that none of their projects failed to satisfy user expectations
- 30% of companies sampled did not respond to this question.

These figures are equally alarming and we speculate that this area particular to software project management methodologies currently in use in Mauritius requires further research.

Factors for Deviation in Project Achievement

Responses for the factors for deviation in project achievement as supplied by the respondents, were assigned values on a scale from 1 to 5 as shown below:

Strongly Agree (SA)	Agree (A)	Neutral (N)	Disagree (D)	Strongly Disagree (SD)
1	2	3	4	5

The mean score was the average value calculated for a particular question. This result then depicts a tendency of the response for a particular question. In figure 7 below, the mean score varied between approximately 2 and 2.5 for all the factors under study, which implies that respondents are of the perceived opinion that all of these factors have to be overcome for software projects to be successful in Mauritius.

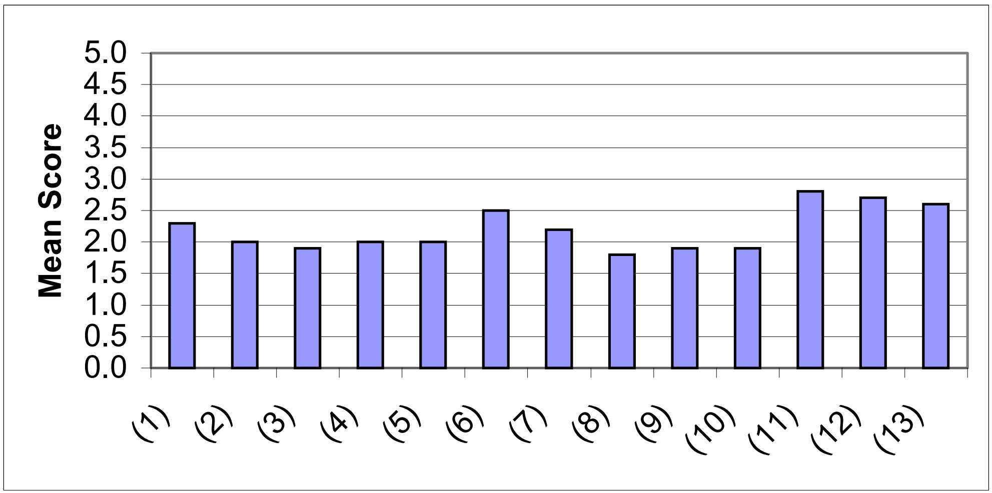


Figure 7. Reasons for deviation.

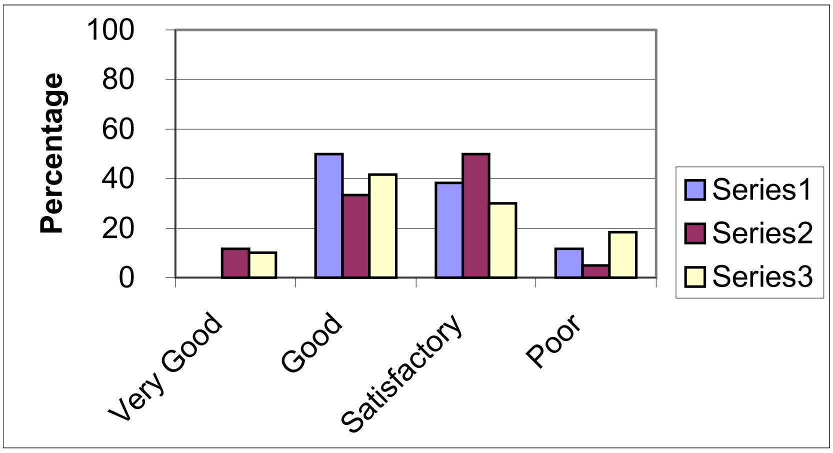
The legends used in figure 7 above is as follows:

- No formal methodology is used
- Poor planning
- User resistance
- Inadequate or improper communication infrastructure (between project team and customer)
- Inappropriate project team
- No executive support
- Failure of project team to understand user needs
- Inadequate or absence of user involvement
- Risk Management
- Difficulty to cope with concurrent projects
- Lack of expertise in innovative/new technology
- Lack of skilled professionals
- Mobility of labour

ANALYSIS OF CLIENT SURVEY RESULTS

The data collected was analysed and represented graphically by considering the percentages as well as mean scores. In the following subsections we will focus only on the data evaluation of the 60 respondents who provided feedback regarding the use of software developed in Mauritius. Note that a 100% response rate was achieved in this part of the survey.

Quality of Software: A User Perspective



Series 1 represents user friendliness of software.
Series 2 represents performance of software.
Series 3 represents security features of software.

Figure 8. Rating of quality.

The majority of users of software are satisfied with the user friendliness features of such software (50% of respondents view this feature as “Good” and 40% as “Satisfactory”). This is probably attributed to the generally short development cycles used in Mauritius, whereby users are exposed early on in the development phase to a prototype version of the software under development.

50% of the respondents are satisfied with the performance of the said software. Respectively 35% and 10% of users view software as “good” and “very good”. Again, this may be attributed to short development cycles, whereby users are exposed early to the software under development.

40% of users rate security features as “Good”, 30% as “Satisfactory” and 10% as “Very Good”. Around 20% of users are of the opinion that security features are poorly addressed. The high degree of user satisfaction in this respect may also be attributed to short development cycles, whereby users are exposed early to the software under development.

Causes of Project Failure to Meet Expected Objectives: A User Perspective

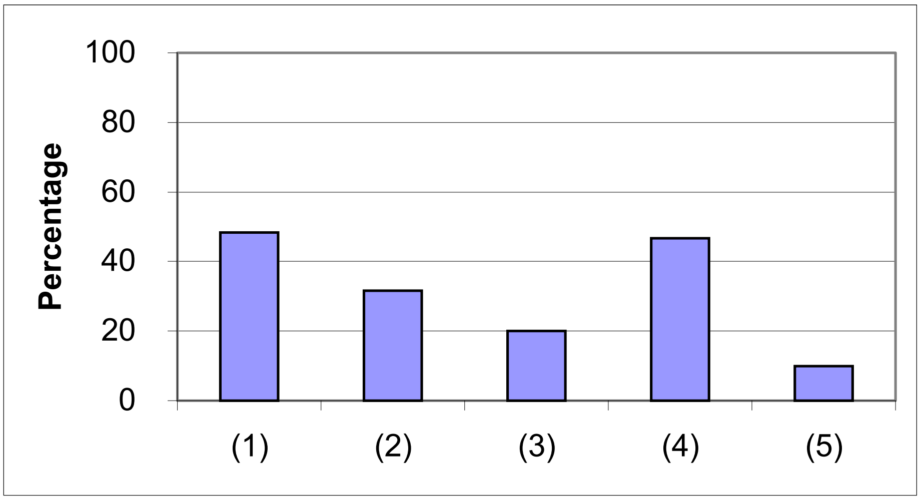


Figure 9. Causes of failure.

The following legend is used in figure 9 above to indicate the causes of project failure to meet expected user objectives:

- lack of communication between supplier and customer
- lack of experience on the part of software developer
- unprofessional mind-set of supplier
- possible lack of best practices for software development and IT management
- none of the above reasons are applicable (i.e. other reasons exist).

Most users are of the opinion that the causes of failure to meet expected user objectives can be attributed to (1) - “lack of communication between supplier and customer” and (4) - “possible lack of best practices for software development and IT management”. One can thus summarise their perception to include improvement of communication, the need to review software development practices by the developers and better IT management is called for.

Factors to Improve User Satisfaction: A User Perspective

The following factors are considered to evaluate customer satisfaction in figure 10:

- improving communication between suppliers and users through regular meetings or informal discussions
- frequent participation of users in the software development process by providing regular feedback
- suppliers should consider user satisfaction seriously rather than just abiding strictly by the specifications agreed upon initially on award of a contract
- a chief information officer (CIO) in charge of issues pertaining to IT should be appointed on the user side
- each user department must recruit its own software developers, IT manager, System Analysts, Database Administrators, etc.
- more effort should be devoted to the elimination of errors in software.

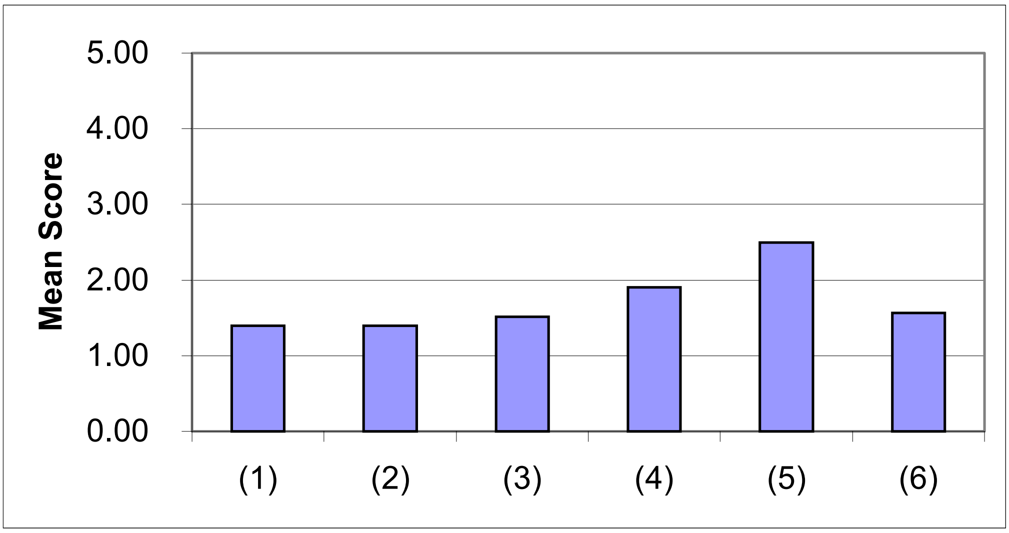


Figure 10. Factors to improve user satisfaction.

The responses for factors that may enhance customer satisfaction with respect to software were assigned values on a scale from 1 to 5 as shown below:

Strongly Agree (SA)	**Agree (A)**	**Neutral (N)**	**Disagree (D)**	**Strongly Disagree (SD)**
1	2	3	4	5

The mean score of the responses lies mostly in the “Agree” to “strongly Agree” category (mean score of between 1.5 and 2) for most of the factors, except for the factor (5). Therefore, from a user’s point of view, most of the factors considered have to be addressed during project management.

CONCLUSION AND FUTURE WORK

Based on the survey that we conducted in Mauritius and reported on in this paper, we conclude that a large number of companies are either not comfortable with existing methodologies, which have western origins, or are not using any methodology at all. Naturally, it is important to use some or other methodology rather than not use any methodology at all. The absence of a methodology advances chaos in a software development process and may contribute to the failure of a project. In such situations software development as well as project management tend to be performed on an ad hoc basis, thereby leading to inefficient management of time, budget and quality.

From the results of our survey, we observe that project management concepts, especially those of western origins, may not be universally applicable, as is demonstrated in the Mauritius case. Developing countries have to constantly face a shortage of skilled staff, difficult economic and social conditions, weak political institutions, as well as deeply rooted cultural and religious beliefs. Therefore, there is a need to encourage the emergence of project management methodologies of a certain indigenous nature, which can cope with the actual status of such countries and stand a better chance of survival (Strucenbruck and Zomorrodian, 1987).

REFERENCES

1. Adler, N. J. (1983). *Cross-cultural management research: the ostrich and the trend*. Academy of management review. 8(2), 226-232.
2. CCTA: Central Computer and Telecommunications Agency. (2002). *Project Management industry initiatives*. Accessed 12.12.2002, online at http://www.ogc.gov.uk/prince/progmgtbrief1997.pdf.
3. Helmerich, A. (2002). *Background and History of the Euromethod Project*. Accessed 31.05.2002, online at http://projekte.fast.de/Euromethod/history.htm.
4. Hofstede G. (1983). *Cultural dimensions for project management*. International Journal of Project Management. 1(1), 41-48.
5. Hughes, B. and Cotterell, M. (2002). *Software Project Management,* 3rd edition. McGraw-Hill: London, UK.
6. Moore, J. (2002). *ISO 12207 and Related Software Life-Cycle Standards*. Accessed 03.09.2002, online at http://www.acm.org/tsc/lifecycle.html.

7. Murithi, N.; Crawford, L. (2003). *Approaches to project management in Africa: implications for international development projects.* International Journal of Project Management. 21(5), 309-319.
8. Nessan, J. (1993). *Developing the African Manager: The good, the bad and the competent*, Management Education and Development. 24(4), 388-394.
9. OGC: Office of Government Commerce. (2002). *Introduction to PRINCE2 – Management Overview*. Accessed 24.05.2002, online at http://www.ogc.gov.uk/prince/about_intro.htm.
10. Ralph, L. K. (1999). *Using Project Management to become ISO 9000 certified*. Information Technology: The Executive's Journal, 18-26.
11. Ralph, L.K. and Ludin, I.S. (1996). *Developing a Project Management Methodology for IS Environments*. Managing System Development. 16(5), 1-4.
12. Stuckenbruck, L. C. and Zomorrodian, A. (1987). *Project management: the promise for developing countries*. International Journal of Project Management. 5(3), 167-175.
13. Turner, J.R. (1993). *The handbook of project-based management: improving the processes for achieving strategic objectives*, McGraw-Hill: London, UK. Cited in Ndiritu Murithi, Lynn Crawford 2003, *Approaches to project management in Africa: implications for international development projects*, International Journal of Project Management, 21(5), 309-319.

An Assessment of Software Project Management Maturity in Mauritius

Aneerav Sukhoo, Andries Barnard, Mariki M. Eloff, and John A. Van der Poll
University of South Africa, Pretoria, South Africa

aneeravsukhoo@yahoo.com barnaa@unisa.ac.za
eloffmm@unisa.ac.za vdpolja@unisa.ac.za

Abstract

It is sometimes very difficult for an organization to adopt a specific software project management methodology in a short space of time. It requires sufficient time, adequate financial support and skilled human resources in order to start with a comprehensive methodology. It is, however, often more appropriate to use a maturity model so as to progress from one maturity level to the next.

Assessment of the maturity level of an organization provides a good benchmark to rate the success of its operations. One such exercise was carried out in South Africa in 2003, and the overall average project management maturity was found to be 2.92 (Sonnekus & Labuschagne, 2004) on a scale of 1 to 5. The maturity level was found to be closely linked to the success rate of projects. In this paper we report on a similar exercise conducted in Mauritius regarding the maturity level of software development projects. The average maturity of software development companies in Mauritius can provide a useful indication of, among others, the current status of software project management with a view of bringing about improvement in this sector.

Given that Mauritian software development companies are making use of European/Western software project management methodologies, this study has been carried out and a preliminary attempt was made to also assess their ability to deal with factors related to cultural, social, economic and political situation within the local context. These factors, when incorporated into existing project management methodologies, can bridge the gap between developing and developed countries and also contribute towards the globalization of software project management.

Keywords: software project management, maturity level, Mauritius, project management methodology, developing country.

Introduction

IT professionals generally have a grasp of the success rate of software projects as compared to other types of projects. The CHAOS report is published regularly to show the percentage of, among others, successfully completed software projects in the USA. The Standish Group (2002) reports that only 34% of projects were completed successfully.

It is sometimes very difficult for an organization to adopt a specific software project management methodology in a short space of time. It requires sufficient time, adequate financial support and skilled human resources in order to start with a comprehensive methodology. A maturity model can allow organizations to progress from one maturity level to the next at their own pace.

Basically, project management maturity, and software project management maturity in particular, is the progressive development of an enterprise-wide project management approach, methodology, strategy, and decision-making process (Project Management Experts, 2004). A number of maturity models has been developed to allow organizations to progress at their own pace and some of these models has been summarized by Schiltz (2003) and Sonnekus and Labuschagne (2004). In a survey carried out in some Swiss organizations, Schiltz (2003) showed that project management maturity could be associated with project success. Such a correlation was also demonstrated by Sonnekus and Labuschagne (2004). Therefore, in an attempt to achieve a higher rate of successfully completed projects, it is worthwhile to consider enhancement of the project maturity level.

Although not all companies use project management maturity models, it is possible to assess their maturity levels. One such framework was put forward by Kwak and Ibbs (2002). An assessment exercise was carried out by Sonnekus and Labuschagne (2004) in the South African context by means of distribution of a questionnaire to various stakeholders. These assessment methodologies provide useful information on the state of project management, particularly software project management, within a country with the intention of bringing about improvements. Furthermore, this survey by Sonnekus and Labuschagne (2004) together with the survey carried out by Schiltz (2003) showed that project management success is linked to a high project maturity level.

A similar exercise as the one conducted by Sonnekus and Labuschagne (2004) in South Africa, was carried out in Mauritius to determine its software project management maturity level. A statistical approach is adopted with respect to the data gathered. Although Mauritius is facing a shortage of skilled labor, various programs have been initiated (Website of Ministry of Information Technology and Telecommunications, 2004), for example:

- a computer proficiency program has been launched to allow the public to benefit from Information Technology (IT) courses at a nominal fee;
- a second university has been set up to offer IT degree courses;
- access to Internet is now provided in most post-offices on the island or in public places;
- a school IT program will be launched to prepare the young generation to master basic IT skills; and
- a loan scheme at a low interest rate has been launched to allow each family to purchase a computer.

A previous survey was conducted to evaluate the current status and use of software project management methodologies in Mauritius (Sukhoo, Barnard, Eloff & Van der Poll, 2004a). This survey showed that necessary steps should be taken to increase the rate of software project success. Furthermore, a substantial number of Mauritian software companies indicated that they were not comfortable with methodologies developed and used exclusively by Western countries that did not account for the economic rationality and cultural differences and the need to cope with political and community demands on the projects' resources in a developing country environment (Sukhoo et al., 2004a). This view is also expressed by Muriithi and Crawford (2003) who argue that software project management concepts are not universally valid due to assumptions on human behavior and cultural difference. Turner (1993) also argues that software project management techniques of western origin are not straightforward procedures that can be learnt and implemented due to cross-cultural problems.

This paper presents an analysis of a survey carried out during 2004 to determine the software project maturity level in the Mauritian context. This survey was conducted by means of a questionnaire based on the one prepared by Sonnekus and Labuschagne (2004). In order to deal with scenarios related to problems in the Mauritian software industry, the questionnaire designed by Sonnekus and Labuschagne (2004), were expanded on in order to capture some specificities of Mauritius. The motivation of the current survey is presented by an earlier survey (Sukhoo et al., 2004a), which showed that the success rate of projects leaves much room for improvement.

A study was also conducted to justify the need for a software project management methodology by emphasizing the importance of strengths and weaknesses inherent in software projects (Sukhoo, Barnard, Eloff & Van der Poll, 2004b). These two studies form part of a larger initiative to develop a software project management methodology specific for Mauritius.

Background on Software Project Management Methodologies

Several project management methodologies are available for software development projects as well as for other types of projects. In the preparation of the questionnaire used for this survey, the Project Management Body of Knowledge (PMBOK® (PMI, 1996)) as well as some project maturity models have been combined to ascertain the maturity level of software project management of organizations. The questionnaire used may be found in appendix B.

PMBOK®

PMBOK®, developed by the Project Management Institute (PMI), is used globally (PMI, 1996) and the current version was released in 2000. The PMBOK® methodology comprises 9 knowledge areas grouped into core functions and facilitating functions as in Figure 1. Each knowledge area is further divided into a number of processes. The total number of processes amounts to 39. These processes are further mapped onto 5 process groups namely initiating, planning, executing, controlling and closing.

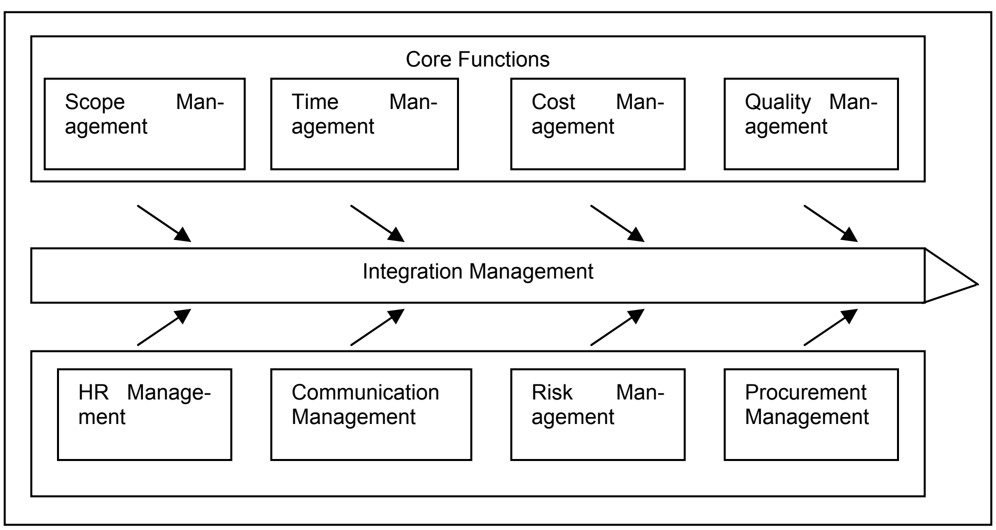


Figure 1: PMBOK® Guide

Project Management Maturity Model (PMMM)

Various PMMMs have been developed (e.g. MicroFrame self-assessment tools as described in Appendix A; Project Management Process Maturity $(PM)^2$ as described in Appendix A; Kerzner's maturity model, as described in Appendix A; and SEI's Capability Maturity Model) and most of them are defined by five maturity levels, 1 through 5 (Schiltz, 2003; Sonnekus & Labuschagne, 2004) as shown in Figure 2.

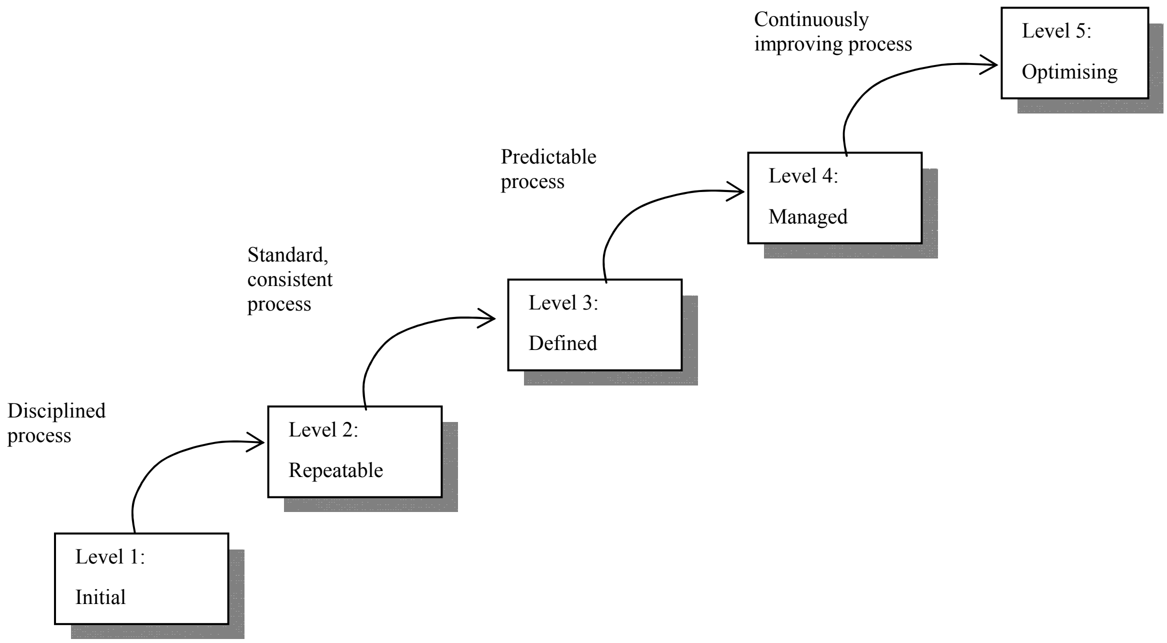


Figure 2: Five levels of maturity model (Paulk, Curtis, Chrissis, & Weber, 1993)

Level 1 – Initial level

The software process is ad-hoc and occasionally chaotic. The success of projects depends on the skills of the project members.

Level 2 – Repeatable level

Processes are repeatable across projects. Basic project management processes are used to keep track of cost, schedule and quality.

Level 3 – Defined level

Project management processes are well established.

Level 4 – Managed level

Quantitative goals for both software products and processes are set by the organization.

Level 5 – Optimizing level

Lessons learnt and defects prevention are considered for continuous process improvement.

The Capability Maturity Model of SEI is a framework that describes the key elements of an effective software development process. The very thorough description of the framework makes it a strong theoretical starting point for developing process maturity models in other areas (such as project management). The process starts with maturity levels for which process capabilities are described. The question that is asked is: "What are the distinguishing capabilities that an organization has when it is at maturity level X?". By describing these capabilities, key process areas are identified, together with the goals that are attained using these process areas. In the next step, common features characterizing the successful implementation of these process areas are determined. Finally, key practices that indicate the successful implementation of the common features, i.e. infrastructure that is in place or activities that are performed, are described by Schiltz (2003) as per Figure 3.

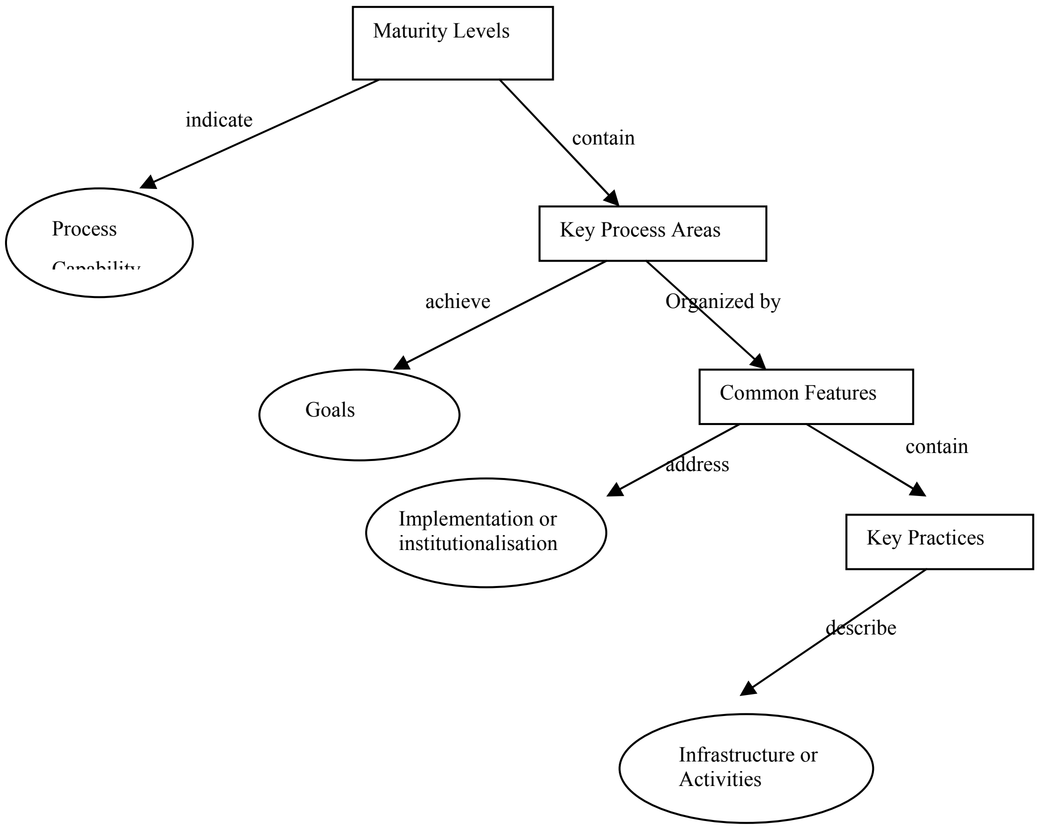


Figure 3: CMM derivation process (Schiltz, 2003)

Research Methodology

Data Collection

Data collection was carried out during 2004 by sending questionnaires via email to 65 companies and individuals actively involved in software development. The companies are those registered in Mauritius and they are listed on the Website of the National Computer Board in Mauritius (Website: http://ncb.intnet.mu). A total of 19 companies responded (i.e. 29%) positively in spite of numerous phone calls made in order to increase the response rate. One of the main reasons identified for the low response rate is that representatives from the companies did not find time to fill in the questionnaire. The 19 respondents include small-sized through medium-sized to large-sized companies as well as some software developers in the IT Units of Ministries. An attempt was made to interview the persons concerned but it failed because these persons could not find the time to contribute to this research.

As mentioned earlier in this paper, this questionnaire was based on work by Sonnekus and Labuschagne (2004). The first section of the questionnaire of Sonnekus and Labuschagne (2004) was adopted as is. Not all questions in the second section were considered. A third section was included to gather information related to economic, cultural and political implications specific to Mauritius.

Analysis of Survey Data

The data collected was input into a Microsoft Excel spreadsheet for analysis.

Perceived Maturity Level versus Observed Maturity Level

The mean values of data from the various companies were used to interpret the maturity levels as follows:

$$M_{perceived} = \frac{\sum_{i=1}^{n} L_i}{n}$$

$$M_{observed} = \frac{\sum_{i=1}^{p} \left[\frac{\sum_{j=1}^{q} S_i}{q} \right]}{p}$$

L = Perceived maturity level

i = i^{th} company/individual

n = maximum number of responses received

S = process maturity level

j = j^{th} process

q = maximum number of processes with maturity levels between 1 and 5 (i.e. excluding responses which are not applicable)

p = maximum number of processes considered.

The results are summarized in Table 1:

Table 1: Average maturity level	
Perceived maturity level	2.26
Observed maturity level	2.29
Difference	0.03

Self-assessment of the perceived maturity level is carried out by judging the appropriateness of the maturity level of an organization with respect to the applicable processes, documentation, management and metrics. Table 1 shows that there is a slight underestimation by companies/individuals in the assessment of maturity levels.

From Table 1, it can be observed that Mauritian software development companies lie at an average maturity level of 2.29 and this possibly indicates an effort to achieve maturity level 3. With the vision of the Government of Mauritius to transform the island into a Cyber Island and to develop Information and Communication Technologies (ICT) as the fifth pillar of the economy (Website of Ministry of Information Technology and Telecomunications, 2004), maturity level 3 appears to be an acceptable level for such ambitious initiatives. It is worth noting that the rapid development in the ICT sector has made people conscious of the use of project management methodologies in software development. Basic processes in the achievement of results are used and sufficiently documented. Basic metrics to track cost, schedule and quality are also used to prevent projects getting out of control and leading to project failures. This is in all probability a good sign for an emerging economy.

Observed Maturity Level by Knowledge Area

The maturity level of each process is evaluated and the average results are summarized in Table 2:

Table 2: Maturity levels by knowledge area

Process Area		Average maturity level
Integration Management		2.28
Core Functions	Scope Management	2.46
	Time Management	2.34
	Cost Management	2.50
	Quality Management	2.35
Average for Core Functions		2.39
Facilitating Functions	Human Resource Management	2.29
	Communications Management	2.25
	Risk Management	1.84
	Procurement Management	2.39
Average for Facilitating Functions		2.19

The above results can be depicted graphically in Figure 4:

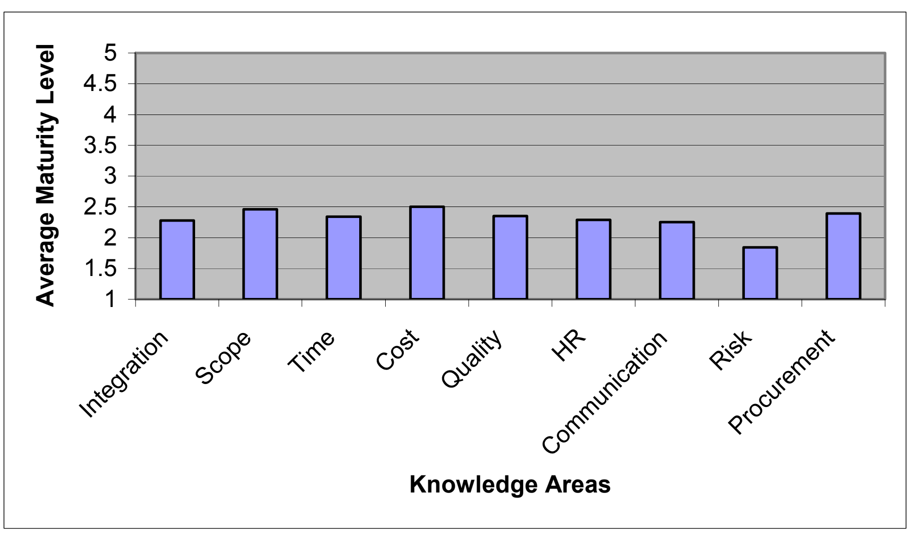


Figure 4: Average Maturity level by Knowledge Area

Most of the knowledge areas score an average maturity level exceeding the value 2, except for Risk Management. This shows that Mauritian software developing companies need to pay more attention to risk management, which may be a major cause of software project failure.

Cost management has achieved the highest maturity level as compared to other knowledge areas. This is an indication that Mauritian software developing companies are quite concerned about cost overruns. Therefore, metrics to track cost of projects are given high importance.

Scope Management is equally treated with high priority given that this is a knowledge area that has to be considered at an early stage. An incorrectly formulated project scope will certainly have major impacts on subsequent phases of the project. Therefore, it is imperative to carefully determine the appropriate scope so as to minimize the risk of the project getting off track during later stages.

Maturity Level by Process Groups

Each knowledge area of PMBOK®, considered in the previous section, is subdivided into a number of processes and these processes are mapped onto the following five process groups:

- initiating
- planning
- executing
- controlling
- closing

A further analysis of the data collected by virtue of the above process groups was carried out. The results are given in Table 3:

Table 3: Average maturity level by process group

Knowledge areas and processes	Initiating	Planning	Executing	Controlling	Closing
Integration Management					
Plan Development		2.33			
Plan Execution			2.33		
Integrated Change Control				2.17	
Scope Management					
Initiation	2.67				
Scope Planning		2.39			
Scope Definition		2.44			
Scope Verification				2.56	
Scope Change Control				2.22	
Time Management					
Activity Definition		2.32			
Activity Sequencing		2.35			
Activity Duration Estimating		2.47			
Schedule Development		2.44			
Schedule Control				2.12	
Cost Management					
Resource Planning		2.65			
Cost Estimating		2.59			
Cost Budgeting		2.53			
Cost Control				2.25	

Knowledge areas and processes	Initiating	Planning	Executing	Controlling	Closing
Quality Management					
Quality Planning		2.25			
Quality Assurance			2.4		
Quality Control				2.4	
Human Resource Management					
Organizational Planning		2.11			
Staff Acquisition		2.44			
Team Development			2.31		
Communications Management					
Communications Planning		2.33			
Information Distribution			2.17		
Performance Reporting				2.21	
Administrative Closure					2.29
Risk Management					
Risk Management Planning		1.92			
Risk Identification		2.00			
Qualitative Risk Analysis		1.69			
Quantitative Risk Analysis		1.69			
Risk Response Planning		1.92			
Risk Monitoring and Control				1.83	
Procurement Management					
Procurement Planning		2.33			
Solicitation Planning		2.28			
Solicitation			2.44		
Source Selection			2.33		
Contract Administration				2.47	
Contract Closeout					2.5
Average for each process group	**2.67**	**2.26**	**2.33**	**2.25**	**2.40**

Maturity Level to Cope with Factors such as Culture, Politics, Climate, Economy, etc.

The perceived maturity level versus observed maturity level to cope with other factors like culture, politics, climate, economy, etc. are summarized in Table 4:

Table 4: Average maturity level	
Perceived maturity level	1.95
Observed maturity level	2.12
Difference	0.17

Given that Mauritian software development companies are making use of project management methodologies based on European/Western countries, this study has been carried out to assess their ability to deal with factors within the local context. The results in the above table show an average maturity level of 2.12, while an underestimated value of 1.95 is generally perceived.

There is an indication that companies are aware that these factors have to be taken, consciously or unconsciously, into consideration while managing software projects. These factors have to be incorporated into project management methodologies through a careful study. This research can possibly bridge the gap that exists between developing and developed countries in the field of software project management. The observed maturity level also shows a possible effort to achieve a higher maturity level. Given that popular software project management methodologies like

PRINCE2 (PRINCE2, 2004), PMBOK® (PMI, 2000), SEI's Capability Maturity Model (CMM Online, 2004), that are widely used are not universally applicable (Muriithi & Crawford, 2003). Mauritian software development companies have had to adjust to the local situation. There is a need to enhance further the means to cope with relevant factors that may affect software projects in Mauritius. Achievement of higher maturity levels in this area may very well bridge the gap that exists between developing countries and Western/European countries. Given that attempts are being made for the globalization of software project management methodologies, it is anticipated that this venture will be worthwhile. In this way, developing economies like that of Mauritius, will remain current.

Conclusion

The maturity level of an organization provides a benchmark for the success of its operation. Sonnekus and Labuschagne (2004) showed the link between maturity level and success rate in software projects undertaken in South Africa by means of a survey conducted. A similar survey conducted in Mauritius revealed the maturity level of software development organizations by knowledge area, process and process group. The results obtained are quite encouraging for Mauritius as a developing country. It shows that the island is making a concerted effort to progress in the software development field. The average maturity level already exceeds level 2 (average maturity of 2.29) and there is a trend to reach the next level given that Mauritius is attracting well-established Indian Companies like Satyam Computers Services Ltd and Infosys Technologies Ltd amongst others.

According to Schiltz (2003) and Sonnekus and Labuschagne (2004), a link exists between project management maturity and project success. We conclude by noting that the current survey together with the previous one supports the need for a methodology that can bridge the existing gap in software project management between Mauritius (or possibly other developing countries) and developed countries in an attempt to globalize the discipline.

References

CMM Online. (2004). The Capability Maturity Model for Software® (SW-CMM®). Retrieved June 15, 2004, from http://www.teraquest.com/cmmOnline/Summary_Ls.html

Kwak, Y. H., & Ibbs, C. W. (2002). Project management process maturity $(PM)^2$ Model. *Journal of Management in Engineering, 18*(3), 150-155.

Muriithi, N., & Crawford, L. (2003). Approaches to project management in Africa: Implications for international development projects. *International Journal of Project Management*, *21*(5), 309-319.

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability maturity model for software, version 1.1. Retrieved October 14, 2004, from http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr24.93.pdf

PMI (1996). *A guide to the project management body of knowledge*. PMI, USA.

PRINCE2 (2003). Retrieved November 19, 2004, from http://www.teraquest.com/cmmOnline/Summary_Ls.html

Project Management Experts, PM Solutions (2004). What is project management maturity? Retrieved July 27, 2004, from http://www.pmsolution.com/maturitymodel/whatispmm.htm

Schiltz, S. J. (2003). *A practical method for assessing the financial benefit of project management.* Master of Science thesis. City University. Retrieved May 19, 2004, from http://www.pmi-switzerland.ch/knowledge/costofbadpm_schiltz_v11.pdf

Sonnekus, R., & Labuschagne, L. (2004). Establishing the relationship between IT project management maturity and IT project success in a South African context. *Proceedings of the 2004 PMSA Global Knowledge Conference,* Johannesburg, South Africa, 183-192.

Sukhoo, A., Barnard, A., Eloff, M.M., & Van der Poll, J.A. (2004a). A survey of project management tools, techniques and methodologies used in Mauritius: The current status. *Proceedings of the 2004 PMSA Global Knowledge Conference,* Johannesburg, South Africa, 320-329.

Sukhoo, A., Barnard, A., Eloff, M.M., & Van der Poll, J.A. (2004b). Towards a framework for evaluating strengths and weaknesses of software projects. *Proceedings of the 2004 PMSA Global Knowledge Conference,* Johannesburg, South Africa, 193-199.

The Standish Group (2002). CHAOS Report as cited by Sonnekus, R., & Labuschagne, L. (2004). Establishing the relationship between IT project management maturity and IT project success in a South African context. *Proceedings of the 2004 PMSA Global Knowledge Conference*, Johannesburg, South Africa, 183-192.

Turner, J.R. (1993). *The handbook of project-based management: Improving the processes of achieving strategic objectives*. London: McGraw-Hill.

Website of Ministry of Information Technology and Telecommunications (2004). Retrieved November 19, 2004, from http://telecomit.gov.mu

Appendix A – Maturity Models

The maturity models described below are as presented by Schiltz (2003).

MicroFrame's Self Assessment Tool

Microframe Technologies, together with Project Management Technologies, have developed and made available on the Internet a self-assessment tool for project management maturity with 50 multiple-choice questions (Enterprise Planning Associates, 2000). The result of this quick self-assessment is a ranking in one of the following five categories:

Level 1 – Ad-hoc: The project management process is described as disorganized, and occasionally even chaotic. Systems and data processes are not defined. Project success depends on individual effort. Chronic cost and schedule problems.

Level 2 – Abbreviated: Some project management processes and systems are established to track cost, schedule, and performance. Underlying disciplines, however, are not well understood or consistently followed. Project success is largely unpredictable and cost and schedule problems are the norm.

Level 3 – Organized: Project management processes and systems are documented, standardized, and integrated into an end-to-end process for the company. Project success is more predictable. Cost and schedule performance is improved.

Level 4 – Managed: Detailed measures of the effectiveness of project management are collected and used by management. The process is understood and controlled. Project success is more uniform. Cost and schedule performance conforms to plan.

Level 5 – Adaptive: Continuous improvement of the project management process is enabled by feedback from the process and from piloting innovative ideas and technologies. Project success is the norm. Cost and schedule performance is continuously improving.

(PM)^2 Maturity Assessment Methodology

A research team formed by Professor William C. Ibbs at the University of California at Berkeley set itself the goal to investigate the financial and organizational benefits to organizations that result from the implementation of project management processes. One of their research steps was to develop a five-level "Project Management Process Maturity" $(PM)^2$ model (see figure below) that would allow them to collect and compare project management process information for a number of organizations to be used in further research studies. The five levels of maturity are described as follows:

Level 1 – Ad-hoc Stage: Organizations at level 1 do not use formal procedures for executing a project. Project activities are poorly defined and cost estimates are inferior.

Level 2 – Planned Stage: At the planned stage, informal and incomplete procedures manage a project. The organization has a strength in doing similar and repeatable work.

Level 3 – Managed Stage: Most of the project management problems are identified and informally documented. PM data for project planning and management are collected across the organization.

Level 4 – Integrated Stage: At the integrated stage, an organization can manage, integrate, and control multiple projects efficiently. PM process data are standardized, collected, and stored.

Level 5 – Sustained Stage: At the sustained stage, PM processes are continuously improved. PM data are collected and rigorously analyzed to improve processes. Innovative ideas are vigorously pursued.

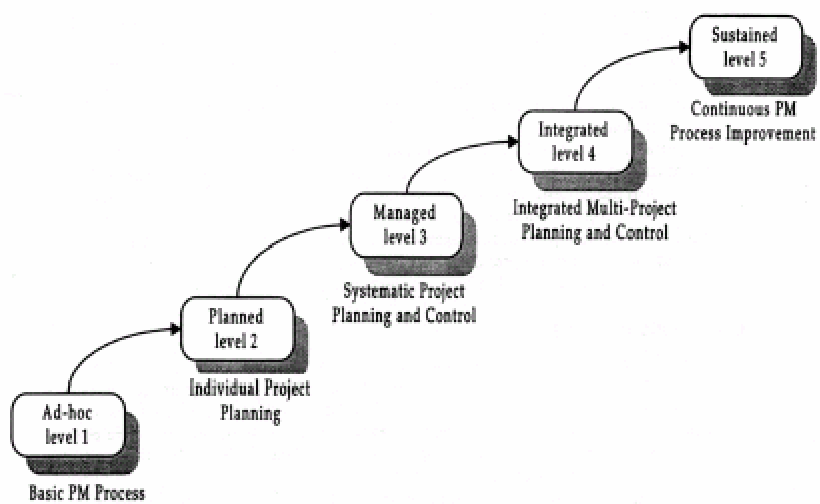


Figure 5: Five levels of $(PM)^2$ (Schiltz, 2003)

Kerzner's Project Management Maturity Model

Harold Kerzner and the International Institute for Learning (IIL) view project management as a core competency that many companies must develop in order to remain competitive in the market. In this context, project management maturity models are important strategic tools for senior management allowing an organization to benchmark its capabilities in respect of project management. As such, a project management maturity assessment model is a tool for establishing project management excellence, which is considered a condition for success.

Like $(PM)^2$ and CMM, Kerzner's maturity model defines five levels by which an organization is ranked from insufficient project management processes to adequate project management processes leading to continuous improvement. These five levels are shown in the figure below and are described as follows:

Level 1 – Common Language: The organization recognizes the importance of project management and the need for a good understanding of the basic knowledge on project management.

Level 2 – Common Processes: At his level, the organization recognizes that common processes need to be defined and developed so that project success can be repeated.

Level 3 – Singular Methodology: The organization defines a single methodology for project management in order to take advantage of the associated synergistic effect.

Level 4 – Benchmarking: The organization recognizes that process improvement is necessary to maintain competitive advantage.

Level 5 – Continuous Improvement: At this level, the organization evaluates the information obtained through benchmarking and decides how to improve its processes.

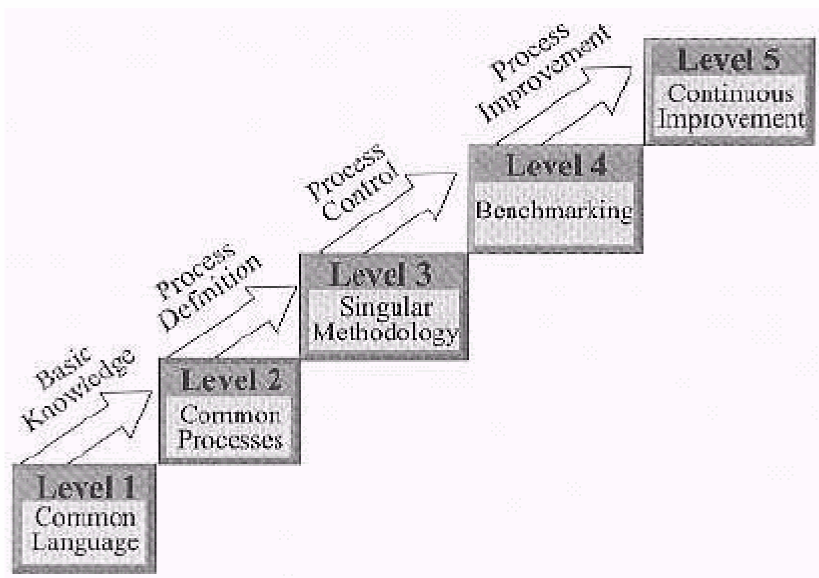


Figure 6: Kerzner's maturity levels (Schiltz, 2003)

Appendix B – Questionnaire Project Management Survey – Maturity vs Project Success Definitions

➢ Please consider the following definitions carefully before completing the questionnaire. The following definitions are referred to the maturity levels in questions in section A and B.

Maturity Level 1 - Initial Process

- **Processes** - No established practices and standards.
- **Documentation** - Loose and ad-hoc.
- **Management** - Management understands the definition of a project, and is aware of the need for project management.
- **Metrics** - Collected informally on an ad-hoc basis.

Maturity Level 2 - Structured Process and Standards

- **Processes** - Processes exist, but are not considered an organizational standard.
- **Documentation** - Documentation exists on the basic processes.
- **Management** - Management supports the implementation of project management, but understanding and involvement is not consistent / applied to all projects. Large projects are executed in a systematic fashion, and management is involved in such projects.
- **Metrics** - Basic metrics to track cost, schedule and technical performance exist.

Maturity Level 3 - Organizational Standards and Institutionalized Process

- **Processes** - All project management processes are in place and established as organizational standards. These processes involve the clients as members of the project team. Nearly all projects use these processes.
- **Documentation** - Documentation exists on all the processes.
- **Management** - Management is regularly involved in input and approval of key decisions.
- **Metrics** - Metrics are formally collected and each project is evaluated and managed in light of other projects.

Maturity Level 4 - Managed Process

- **Processes** - project management processes, standards and supporting systems are integrated with other corporate processes and systems.
- **Documentation** - Processes and standards are documented to support using metrics to make project decisions.
- **Management** - Management understands its role in the project management process. There are different management styles and project management requirements for different projects.
- **Metrics** - Efficiency and effectiveness metrics are used. All projects, changes and issues are evaluated based upon metrics from cost estimates, baseline estimates, and earned value calculations.

Maturity Level 5 - Optimizing Process

- **Processes** - Processes are in place and actively used to improve project management activities.
- **Documentation** - Lessons learned are regularly examined and used to improve project management processes, standards and documentation.
- **Management** - Management is focused not only on effectively managing projects but also on continuous improvement.
- **Metrics** - The metrics collected during project execution are used to understand the performance of a project and to assist in the making of organizational management decisions for the future.

Section A
Project Management Maturity

- Please rate each of the following statements according to the maturity levels on the previous page, by making an **X** in the appropriate box. If your organization does not implement a specific section, please mark the N/A (not applicable) box.

1. Overall Level of Maturity	**1**	**2**	**3**	**4**	**5**
a. What do you think the overall level of project management maturity is in your organization?					

2. Project Integration Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. Project Plan Development Integrating and coordinating all project plans to create a consistent, coherent document.						
b. Project Plan Execution Executing the project plan by performing the activities included therein.						
c. Integrated Change Control Coordinating changes across the entire project.						

3. Project Scope Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. Initiation Authorizing the project or phase.						
b. Scope Planning Developing a written scope statement as the basis for future project decisions.						
c. Scope Definition Subdividing the major project deliverables into smaller, more manageable components.						
d. Scope Verification Formalizing acceptance of the project scope.						
e. Scope Change Control Controlling changes to project scope.						

4. Project Time Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. Activity Definition Identifying the specific activities that must be performed to produce various project deliverables.						
b. Activity Sequencing Identifying and documenting interactivity dependencies.						
c. Activity Duration Estimating Estimating the number of work periods that will be required to complete individual activities.						
d. Schedule Development Analyzing activity sequences, activity durations and resource						

	N/A	1	2	3	4	5
requirements to create the project schedule.						
e. Schedule Control Controlling changes to the project schedule.						

5. Project Cost Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. Resource Planning Determining what resources and what quantities of each should be used to perform project activities.						
b. Cost Estimating Developing an estimate of the costs of the resources required to complete project activities.						
c. Cost Budgeting Allocating the overall cost estimate to individual work activities.						
d. Cost Control Controlling changes to the project budget.						

6. Project Quality Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. Quality Planning Identifying which quality standards are relevant to the project and determining how to satisfy them.						
b. Quality Assurance Evaluating overall project performance on a regular basis to provide confidence that the project will satisfy the relevant quality standards.						
c. Quality Control Monitoring specific project results to determine if they comply with relevant quality standards and identifying ways to eliminate causes of unsatisfactory performance.						

7. Project Human Resource Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. Organizational Planning Identifying, documenting and assigning project roles, responsibilities and reporting relationships.						
b. Staff Acquisition Procuring the required human resources and assigning it to the project.						
c. Team Development Developing individual and group competencies to enhance project performance.						

8. Project Communications Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. Communications Planning Determining the information and communications needs of the stakeholders.						
b. Information Distribution						

Making required information available to project stakeholders in a timely manner.						
c. **Performance Reporting** Collecting and disseminating performance information. This includes status reporting, progress measurement, and forecasting.						
d. **Administrative Closure** Generating, gathering, and disseminating information to formalize a phase or project completion.						

9. Project Risk Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. **Risk Management Planning** Deciding how to approach and plan the risk management activities for a project.						
b. **Risk Identification** Determining which risks might affect the project and documenting their characteristics.						
c. **Qualitative Risk Analysis** Performing a qualitative analysis of risks and conditions to prioritize their effects on project objectives.						
d. **Quantitative Risk Analysis** Measuring the probability and consequences of risks and estimating their implications for project objectives.						
e. **Risk Response Planning** Developing procedures and techniques to enhance opportunities and reduce threats to the project's activities.						
f. **Risk Monitoring and Control** Monitoring residual risks, identifying new risks, executing risk reduction plans, and evaluating their effectiveness throughout the project life cycle.						

10. Project Procurement Management	**N/A**	**1**	**2**	**3**	**4**	**5**
a. **Procurement Planning** Determining what to procure and when.						
b. **Solicitation Planning** Documenting product requirements and identifying potential sources.						
c. **Solicitation** Obtaining quotations, bids, offers, or proposals, as appropriate.						
d. **Source Selection** Choosing from among potential sellers.						
e. **Contract Administration** Managing the relationship with the seller.						
f. **Contract Closeout** Completion and settlement of the contract, including resolution of any open items.						

Section B
Maturity to Cope with Other Factors

1. **Overall Level of Maturity to cope with other Factors (e.g. cultural, political, climatic and economic)**	**1**	**2**	**3**	**4**	**5**
a. What do you think about the overall level of maturity to cope with factors like culture, politics, religion, infrastructure, bad climatic conditions, etc. that affect projects in your organization?					

2. **Other aspects to be considered during Project Management**	**N/A**	**1**	**2**	**3**	**4**	**5**
a. Organizational Culture Existence of an organizational culture to keep project team integrated as a community to achieve corporate goal.						
b. Individualism/collectivism Ability to cope with individualist/collectivist nature of staff. (Individualistic nature: where ties with others, including families, are loose and therefore staff may view that their work is more important than others.) (Collectivist nature: where ties with others, including families, are strong and therefore others are viewed with greater values)						
c. Uncertainty avoidance Coping with uncertainty avoidance. (Uncertainty avoidance: The extent to which low risk taking and emotional resistance to change are preferred)						
d. Gender equality Ability to cope with gender issues and to treat both as equal during recruitment or other circumstances.						
e. Authority Managing authority so that decisions are not delayed. (High authority represents a high acceptance by less powerful individuals. It is often associated with taller organizational structures and unwillingness to make decisions lower down without reference to superiors)						
f. Religion Ability to cope with religious dimension so that projects are not affected. For instance, during certain religious festivals (like Maha Shivratree, Eid, Easter, Spring festival, etc.), some employees may take some days of leave that may impact on projects.						
g. Infrastructure Ability to cope with infrastructure problems (e.g. transport, education system, new technology, communications infrastructure, etc.) that may have impacts of projects.						
h. Climate Ability to cope with delays due to natural calamities (like cyclone and heavy continuous rain).						
i. Politics Ability to cope with politics that may have influence on the						

	way projects are managed.						
j.	**Legal** Ability to cope with legal shortcomings (e.g. inadequate protection against software piracy, etc).						
k.	**Economic** Ability to cope with economic situation (e.g. the bad economic situation may favor a degradation in software quality or highly skilled professionals cannot be attracted for software development).						

Section C
Project Success and Failure

Project success

A successful project is a project that is delivered on time, within the budget, and on brief (in scope).

Project failure

A failed project is a project that is never completed or does not meet customer expectations.

Project challenge

A challenged project is a project that is completed, but is either late, over-budget, or does not meet all the requirements.

1.	**Total** number of projects completed in last 12 months by your organization	
2.	Number of **successful** projects in last 12 months	
3.	Number of **failed** projects in last 12 months.	
4.	Number of **challenged** projects in last 12 months	

Section D

General Information

1. Total no. of staff in your organization (IT and non-IT) ______________________

Optional information

2. Name of company ____
3. Position of respondent ____
4. Would you like to receive an electronic copy of the survey results Yes/No (delete whichever is not applicable)

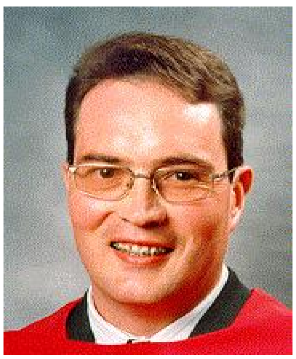
   If yes, please provide your email/postal address

   ____

Biographies

Aneerav Sukhoo, is a project manager working for the Ministry of Information Technology & Telecommunications in Mauritius. He has 15 years of experience, out of which 8 years in IT project management, in dealing with implementation of IT projects in the Public Service.

He has a Honours Bachelor degree in computer Science from the University of South Africa and a Master of Information Technology degree from Charles Sturt University. He is presently preparing a PhD thesis at the University of South Africa for the development of a software project management methodology.



Andries Barnard, associate professor in the Department of Computer Science and Information Systems, holds a PhD (Computer Science). He teaches undergraduate courses in automata theory and formal languages and project management, as well as postgraduate courses in project management and research methodology. His research interests include computer ethics and graph grammar languages.

Mariki Eloff received a PhD Computer Science degree from the Rand Afrikaans University (RAU), South Africa. She has presented research papers at international and national conferences focusing on information security. She joined Unisa as an associate professor in the School of Computing during October 2002. She has assisted in the organization and management of international conferences in information security and has been part of the ISSA Conferences since its inception in 2000.

During 2001 she participated in information security management research projects under the joint leadership of RAU and the South African Bureau of Standards (SABS). She has also been engaged in the development of various information security-training modules for industry.

John A. Van der Poll is an associate professor and acting head of the Department of Systems Organisation in the School of Computing at the University of South Africa (UNISA). He holds a PhD in Computer Science and teaches an undergraduate course in operating systems, as well as postgraduate courses in compiler construction and formal program verification. His research interests are in formal specification languages and automated reasoning.

Accommodating Soft Skills in Software Project Management

Aneerav Sukhoo, Andries Barnard, Mariki M. Eloff, John A. Van der Poll
University of South Africa, Pretoria, South Africa

aneeravsukhoo@yahoo.com barnaa@unisa.ac.za
eloffmm@unisa.ac.za vdpolja@unisa.ac.za

Mahendrenath Motah
University of Technology, Mauritius

mmotah@utm.intnet.mu

Abstract

Software project management probably is a sector that has witnessed the highest rate of project failure in the world. This is not the case with project management concerned with other disciplines due to better management of their inherent strengths and weaknesses.

Hard skills, often described as a science and comprising processes, tools and techniques applied to projects are the main focus of many project management methodologies. Extensive research has been carried out in the project management domain with greater focus on hard skills. Soft skills described as an art, are very often ignored during software project management. Such skills, acquired through experience, are concerned with managing and working with people, ensuring customer satisfaction and creating a conducive environment for the project team to deliver high quality products within budget and on time and exceeding stakeholder expectations.

This paper summarizes the soft skills that can possibly improve the success rate of software projects. These projects, if not managed properly can lead to escalation of budget and time schedules beyond expectations. Deterioration of quality may be inevitable while deploying efforts to deal with cost overrun and schedule extension.

Research in the field of soft skills required has been carried out at the University of Arkansas. The results show that IT organizations are conscious of the need for soft skills. Furthermore, a number of organizations, like Polaris, Sun Microsystems and Mastek have found it necessary to include soft skills in their training agenda.

Keywords: soft skills, hard skills, IT project management, software project management, project management methodology, emotional intelligence, psychology.

Introduction

Information Technology (IT) project management (software project management) is a sector that has arguably witnessed the highest rate of project failure in the world. Dorsey (2000) pointed out that large information systems projects have been reported to be subject to failure rates between 50-80% and Bupa (2005) stated that according to a recent report by the Standish Group, only one in three IT projects were delivered on time, within budget and according to specification. This is also confirmed by the CHAOS report (The Standish Group, 2001), which is published annually. This is generally not the case with project management concerned with other disciplines due to better management of inherent strengths and weaknesses. IT projects, particularly those of a software nature, have different strengths and weaknesses as compared to traditional engineering projects (Sukhoo, Barnard, Eloff & Van der Poll, 2004a). For instance, some strengths associated with software projects include flexibility, ease of creating backups, scalability, replication and reusability of components while some weaknesses include invisibility, complexity, difficulty to add people to delayed projects and the need for regular upgrades.

However, hard skills remain the traditional main focus of most IT project management methodologies. Hard skills, often described as a science (Belzer, 2004), comprise processes, tools and techniques applied to projects. In managing software projects, tools and techniques related to hard skills are given much attention in an attempt to drive projects towards success. Unfortunately, we find that many software projects do not live up to expectations.

Soft skills, often described as an art (Belzer, 2004), have been identified as critical for project success. They are often concerned with managing and working with people (Kirsch, 2004). These skills are typically acquired through experience (Belzer, 2004). Companies, like Mastek, Polaris and Sun Microsystems, being conscious of the importance of soft skills, have incorporated such skills into their training agenda (Arora, 2003).

This paper presents the soft skills that can possibly lead to an improvement of the success rate of software projects. These software projects, if not managed properly can lead to an escalation of budget and time schedules beyond expectations. Deterioration of quality may be inevitable while deploying efforts to deal with cost overruns and schedule extensions.

Motivation for this Research

Every year, the CHAOS chronicle (see for example The Standish Group, 2001) reports on the failure rate of IT projects in the USA. Although there was an observed decline in the percentage of failed projects in the USA since 1994 as summarized by Sonnekus and Labuschagne (2004), the same trend may not have been observed in other countries, especially those with a developing economy.

Developed countries like the UK and USA are facing an advantageous situation with the use of well-adapted methodologies, tools and techniques through rigorous research and development initiatives both from academia as well as from professional bodies like the Project Management Institute (PMI) and Office of Government Commerce (OGC). According to a survey carried out by Sonnekus and Labuschagne (2004), the failure rate of IT projects in South Africa was found to be 22%, which can be observed to be comparable to that of the CHAOS chronicle released in 2000 (see Table 1).

Table 1 – USA success rate

Year	Percentage of failed projects
CHAOS - 1994	31%
CHAOS - 1996	40%
CHAOS - 1998	28%
CHAOS - 2000	23%
CHAOS - 2002	15%

Following a survey carried out in 2003 in Mauritius (Sukhoo, Barnard, Eloff & Van der Poll, 2004b), at least 50% of software projects that suffered due to deadline problem, budget overruns and quality problems were found to be 50%, 30% and 10% respectively.

In contrast, traditional engineering projects generally achieve much higher success rates. This is due to exploitation of inherent strengths and prudent management of weaknesses associated with engineering projects that are different from those of software projects (Sukhoo, Barnard, Eloff & Van der Poll, 2004a). The strengths and, in particular the weaknesses associated with engineering projects have been addressed better due to the fact that these projects were managed for many decades and are also better understood. Management of software projects in particular has been carried out for far less than what is the case with traditional projects. In addition to hard skills required, it has been observed that a good mix with soft skills is required for IT organizations (Arora, 2003). When popular project management methodologies like the PMBOK® (Project Management Body of Knowledge) and PRINCE 2 (PRojects IN Controlled Environment) are analyzed closely, one finds that not much emphasis is laid on soft skills.

Furthermore, a survey of soft skills required in IT organizations was carried out at the University of Arkansas (Hathaway, 1999) and the results obtained showed highly desired skills to be problem solving process, listening skills, teamwork, adaptability to new technology and new languages, transferring knowledge to application, time management, visualization and conceptualization skills and verbal communication. Desired skills included the ability to multi-task, dealing with business culture, inter-team communication, interpersonal skills, constructive criticism, organisation skills, stress management and general writing skills. According to the survey, less desired skills comprise leadership, technical writing, dealing with diversification (different cultures) and presentation skills.

Hard Skills

Hard skills are the technical skills required within the confines of a domain. They encompass the following dimensions:

- Processes
- Tools
- Techniques

Aspects pertaining to some of the above may be learnt largely through a formal project management course or from textbooks.

Extensive research has been conducted in project management with greater emphasis on hard skills (Might & Fisher, 1985; Keil, 1995 as cited in Kirsch, 2004). These skills, although of crucial importance, are to be considered along with the broader soft skills according to Belzer (2004) and Moreira (2004). Most project management methodologies show a greater coverage of hard skills. It is observed that research efforts are expended towards automating the process of management of projects with regards to hard skills.

However, Moreira (2004) claims that success in the management of software projects does not rely entirely on hard skills. A combination of hard skills and soft skills may effectively and efficiently steer projects towards success. This is confirmed by the fact that the best jobs are often offered to seasoned professionals who have a good combination of technical knowledge, general business skills and communication skills (Moreira, 2004).

Given that software projects have inherent complexity and invisibility features (Hughes & Cotterell, 2002) that may hamper progress during development stages, it becomes necessary to articulate hard skills together with soft skills.

Soft Skills

Soft skills are the non-technical skills that are often not given due consideration while managing projects. Soft skills, often described as an art (Belzer, 2004), is concerned with managing and working with people, ensuring customer satisfaction with the intention of retaining them and creating a conducive environment for the project team to deliver high quality products within budget and on time and exceeding stakeholders expectations. Soft skills are acquired through experience, but sensitization of students is necessary during formal software project management courses. In addition, psychology may be incorporated into software project management curricula and this is confirmed by Cook (2004) who stated the need for project managers to have a clear grasp of social psychology to insure optimum performance of project teams. According to Taylor (as cited in Heikkila, 2002), one has to see beyond technical capabilities and recognize the value of business-focused skills in order to successfully oversee and manage an IT department.

Soft Skills Dimensions

A plethora of soft skills dimensions has been identified for the management of projects (Belzer, 2004; Moreira, 2004). These soft skills include, amongst others, communication skills, team building skills, flexibility and creativity skills, leadership skills and the ability to manage stress and conflict. Moreover, a number of research initiatives are underway to sensitize stakeholders and project managers regarding the importance of such skills to achieve smooth execution of projects. Software projects, in particular, warrant a careful selection of project managers to achieve project objectives.

Mazarr (2000) mentioned that human beings know more that just “rote, mechanical operations and endless repetition of tasks” and he observed that the best workplaces are filled not by those with the most intimidating technical knowledge but rather by those with the best soft skills.

Communication skills

Communication skills refer to the ability to convey ideas easily and clearly in order to ensure that the team moves towards a common goal (Belzer, 2004). Effective communication is a very important soft skill required by software project managers. According to Belzer (2004), “project managers must have the ability to convey complex ideas easily, clearly articulate what must be accomplished, keep the team moving towards a common goal, foster an environment that allows team members to communicate openly and honestly, admit their own mistakes without losing respect, negotiate, listen and facilitate”. It is generally agreed that effective communication is the key to the success of any endeavor in which people are involved. The IT field is no exception, especially where software development is concerned. Very often, aspects such as language, learning styles, communication networks, preferred learning mode, memory access and retrieval systems, attention and retention, and various other factors affecting learning; information processing and communication are simply disregarded. Particularly, within the ambit of the IT domain, these aspects are minimized in favor of technology (Newstrom & Davis, 2001).

Many managers have now started to accept and appreciate the importance of NLP (Neuro Linguistic Programming) in the learning process, and in establishing effective communication patterns in, and between, organizations. One cannot deny the fact that the success of any project rests upon the judicious management of all information. Hence, the importance of a proper communication network, and various related aspects should be carefully considered, to ensure the success of projects where people are involved (Newstrom & Davis, 2001).

Throughout the software development process, communication of project information should be encouraged. Consideration has to be given to who needs what information, in what format and when (Bruce and Langdon, 2000). Two-way communication must be enhanced, thereby, inviting important feedbacks to keep projects moving in the right direction. In this way, the software project manager may also encourage the project team to be open and honest. The software project manager must ensure that team members are not intimidated by fear for repercussion (Bruce & Langdon, 2000).

Various means of communication like formal meetings, informal discussions and reports etc., must be encouraged. Also, the use of email may be an extremely useful time-saving device and the use thereof should be considered.

Team building

Team building involves the setting up of a team with an appropriate mix of skills to ensure the successful completion of a project. Conflicting behavior between team members should be detected as early as possible. Software project management is typically a team activity, and it is groups of people working together formally that can guarantee the achievement of defined objectives related to software project management. Teamwork is very significant to project management as there is a clear need for considerable interaction between people carrying out different functions, but with a common purpose.

Team work is important at various levels – from senior management level down to and including lower levels. Different types of teams are necessary: inter-disciplinary teams, informal teams, autonomous teams, and others, during the various phases of a project. The collection of individual skills working towards a common goal constitutes a powerful force (Phillips, 2002). The effect is that "The whole is greater than the sum of the parts", as Gray & Larson (2000) rightly pointed out.

Hence, team building and team development will be the manager's responsibility. The project manager will have to ensure and encourage individuals to join teams and create situations to promote effective teamwork. Once in place, the teams should be involved in team processes and carry out their tasks and maintenance functions (Newstrom & Davis, 2001).

Team building helps establish group norms, group behavior and desired attitudes and group identity. Software project managers should ensure that a full complement of people with a commensurate level and mix of skills and personalities are adequately represented on a project (Bruce & Langdon, 2000). Software project managers may seize the opportunity of project review meetings to strengthen teamwork and help build team confidence.

Flexibility and creativity

Since every software project has unique characteristics, each may call for a different combination of components, templates, tools and techniques (Belzer, 2004). The software project manager must also adapt to the needs of the project. Flexibility and creativity can be promoted through the understanding of the personality of individual members, and acceptance of individual preferences (Santrock, 2000).

The use of toolkits and other equipment is possible through an appreciation of the intellectual abilities of the team members and the belief that each and every individual is perfectible. Opportunities provided by the software project leader to the team will help members to initiate actions and encourage creativity.

Leadership

According to Dobbins and Pettman (1997), leadership is the ability to motivate people to work towards achieving common goals, to make ordinary people display extraordinary performance. In short, leadership has been related to a person's skills, abilities and degree of influence to get people moving in a direction, making decisions and doing things that they would normally not have embarked on. Leading is one of the functions of managers and as such software project managers must lead a software project team. Different styles of leadership, some better than others in certain circumstances, are adopted by software project managers. Good teamwork may result from consensus building between project managers and team members rather than adopting a dictatorial style. A software project manager should be someone whom people want to seek out rather than avoid (Bruce & Langdon, 2000). According to Bruce & Langdon (2000), four leadership styles (as per table 2 below) may be associated with software project managers.

Table 2 – Leadership styles

Leadership style	Description
Dictatorial	This style refers to the decision-making process by the project manager alone. The latter displays autocratic and controlling characters. This style may be adopted when the project faces a crisis and when there is no time for consultation between the project manager and individuals concerned with the project. The dictatorial style must be used with care and as infrequently as possible since it demotivates the team members, thereby discouraging teamwork.
Analytic	The project manager reaches a decision by gathering and analyzing facts. This style, which requires good analytical skills, is adapted when a project is running short of time and the correct decision must be taken quickly.
Opinion-seeking	This style involves gathering team members' opinions in order to reach a decision. Team confidence is built and the project manager indicates that the views of the team are valued positively.
Democratic	This style refers to team participation encouragement and involvement in the decision-making process. It motivates the team and helps strengthen their commitment to the project.

Recent work performed in the domain of leadership highlights the importance of emotional intelligence as one of the basic requirements of the modern leader (Goleman, 1997). The behavior of human beings is influenced by emotions since the dawn of time. Many theorists have emphasized the importance of emotions and how to deal with emotional employees and it is nowadays widely accepted that the management of emotions, particularly anger, can assist in stress management. Goleman (1998b) argues that stress can be considered as the most damaging and costly illness in most organizations.

The psychological implications in the choice of an appropriate leadership style cannot be overlooked. The contingency leadership style, considered as the most popular among employees is, however, not generally acceptable to all managers. Many prefer to adopt one particular style and rarely accept change, usually to the detriment of the employee.

Emotional intelligence, leadership styles and human resource management are some of the key issues that need to be considered as desirable soft skills indispensable for successful IT project management.

Organizational effectiveness

To organize is the management function that focuses on arranging and allocating work, authority and resources among an organization's members so that plans may be successfully carried out (University of Mauritius, 2004). Effective organizations are learning organizations. A learning organization takes an institutional permanence, and acceptable modes of behavior become self-evident to its members. Such shared meaning held by members of the organization contributes to the establishment and reinforcement of organizational culture. With the advent of IT, several characteristics that capture the essence of organizational culture have become more evident. As such, aspects such as innovation and risk taking, attention to detail, people orientation, team orientation, outcome orientation, stability, positive aggressiveness and other dimensions which directly or indirectly affect individual members should be given due consideration (Farago & Skyrme, 1995).

One should be cognizant of the fact that organizational culture has several important functions. An organization with a dominant culture defines the boundary which limits the behavior of its members; it conveys a sense of identity among its members; it facilitates the generation of commitment; it enhances internal social system stability and helps hold the organization together; it also serves as a sense-making and control mechanism; and it guides and shapes the attitude and behavior of the members of the organization (Farago & Skyrme, 1995). It is therefore important for software project managers to understand the corporate culture, the organizational dynamics and the individual they are dealing with (Belzer, 2004).

Stress management

Stress refers to pressure, tension or worry resulting from problems (Oxford Advanced Learner's Dictionary, 1999). Stress management is necessary for the software project manager who has to perform under pressure quite often and must not be overwhelmed to the extent that the team is also affected by their behavior. A certain level of stress is important for the software project manager to perform well and this type of stress is called eustress, which is good stress. When stress becomes excessive, it can cause harm to oneself or to the immediate surroundings. Beyond a certain level, it can cause distress.

Distress can be harmful to the extent that it can lead to the following problems that may affect one's health and at the same time hamper progress of the project:

- Physical problems like pain, tiredness, colds, insomnia, dizziness, nausea, palpitations and sweating.
- Emotional problems like anxiety, guilt and lack of control, anger, depression, pessimism, irritability, irrational fears and resentment.
- Mental problems like poor memory and concentration, negative thoughts, confusion and loss of sense of humor. Stress sufferers can lose up to 50% of their aptitude to perform their jobs (Chapman, 2001).
- Behavioral problems like smoking, crying, fidgeting, alcoholism, lashing out and shouting.

Software development is an area where constraints of deadlines, degrading quality of work and budget overrun can rapidly lead to a degeneration of the working environment. Distress will definitely cause a deterioration of the situation. Maintaining a good level of eustress as early as possible and ensuring that it does not turn into distress can help to improve the progress of the project.

Time management

According to PMI (2000), time management includes the processes required to ensure timely completion of projects. The soft skills side of time management refers to the ability of the software project manager to plan, delegate, organize, direct, and control.

These attributes can be improved through experience. A list of obstacles preventing effective time management has to be prepared (a so-called list of time wasters) and ways to eliminate these time wasters should be established and applied.

Change management

Changes are inevitable during the execution of a project and some circumstances may cause the team to feel demotivated. It is, therefore, necessary to adjust the plan accordingly to accommodate changes. Some changes are within the control of the software project manager (like shortening the time schedule) while others are not (like the transfer of one key team member to another project by senior management). In this situation, an assessment of the impact must be made with regards to changes to schedules, budgets and resources involved. Any alteration should be identified and evaluated before committing changes (Proehl, 2001).

One important change management issue is dealing with user resistance. Especially, in the field of software projects, changes are inseparable features as new developments in technology are brought about continually. Both users and team members have to be informed as early as possible. People's concerns need to be taken into consideration and actions to dissipate their fears have to be taken as expeditiously as possible (Proehl, 2001).

In particular, user resistance can cause a software project to fail at the last stage (operational stage) after months or years of assiduous effort. Therefore, this issue has to be taken care of from the outset of the project. One solution is to involve the users throughout the different software project phases (James, 1999).

Trustworthiness

Trustworthiness is the value given to the belief that one can rely on the goodness, strength and ability of someone else (Oxford Advanced Learner's dictionary, 1999). Trust is the basic requirement that can ensure the success of any human enterprise. Trust must prevail at all times between partners and between the manager and his/her subordinates. All relationships ought to be based on trust.

Mutual trust must prevail throughout the project execution between the software project manager and all of the stakeholders involved in the project. Simply meeting deadlines is just one facet of the objectives of a project; a software project manager must also be able to convey that he/she can always be trusted to do what is right at the right time to render the project successful and the client satisfied (Lussier, 1990).

Conflict management

Conflicts are serious disagreements that emerge within the project team during the execution of a software project. They may arise due to disagreements over priorities, allocation of resources, quality of work (Gray & Larson, 2000) or other reasons. Conflicts may be desirable (especially, when they clear ambiguities or when they raise concerns about a certain problem or pitfall) or can hinder the performance of a project (e.g. when both parties in conflict are wrong in the debate concerned). Desirable conflicts are often referred to as functional conflicts while undesirable conflicts are also called dysfunctional conflicts.

Conflict resolution skills are necessary for a software project manager to prevent behaviors of team members from degenerating into irreconcilable situations that may prohibit them from ever working together productively again (Newstrom & Davis, 2001).

Dysfunctional conflict resolution is a challenging task for software project managers. It might not be identified until it becomes too late to overcome. According to Gray & Larson (2000), the following strategies can be employed to manage it:

- *Mediation of conflict*

  This strategy relies on convincing the parties concerned about the need for making concessions in the interest of the project. The project manager has to negotiate the resolution by using reasoning, persuasion and suggesting alternatives.

- *Arbitration of conflict*

  A solution is sought that allows each party to save face.

- *Control of conflict*

  The conflict is controlled by reducing its intensity either by smoothing over the differences or postponing the resolution to a more suitable period after both parties regained composure. In case no solution can be found, project assignments may have to be rearranged so that the persons concerned are not allowed to interact any further.

- *Accepting conflict*

  Sometimes the software project manager may have to accept the conflict through the life of the project but must introduce distractions to ignore the issue.

- *Elimination of conflict*

  Conflicts may escalate to the point that they become intolerable and unacceptable. In such cases, the person at fault must be removed from the project. Software project managers must constantly watch out for behaviors within the project team that may escalate into conflict. The ability to read body language to identify unspoken disagreements may be an important skill for project managers.

Integrating Soft Skills and Hard Skills in Software Project Management

The high failure rate of software projects necessitates actions to be taken as soon as possible in order to improve this discipline. In this context, it is imperative to integrate soft skills along with hard skills in existing and emerging software project management methodologies.

The previous sections provide details about some important soft skills required by software project managers. More soft skills attributes for project managers are embodied in the emotional intelligence framework put forward by Goleman (1998a) and recapitulated by Barry and Du Plessis (2004).

Soft skills attributes of the software project manager that require a high level of activation during the different phases of a project have to be defined and then considered at the right time. A table of activation of soft skills, considered in the previous sections, for different phases of software projects may be prepared as shown in Table 3:

Table 3 – Soft skills activation

Soft Skills	Project management phases				
	Initiating	Planning	Executing	Controlling	Closing
Communication skills			√	√	√
Team building			√		
Flexibility and creativity	√	√	√		
Leadership			√	√	

Soft Skills	Project management phases				
	Initiating	Planning	Executing	Controlling	Closing
Organizational effectiveness		√	√		
Stress Management			√	√	
Time Management		√	√		
Change Management			√	√	
Trustworthiness			√	√	
Conflict Management			√	√	

Table 3 was compiled by means of interviews conducted with project managers/systems analysts of various software development organizations. However, in-depth research is required so that the appropriate soft skills may be added and other skills excluded depending on the circumstances prevailing in an organization, culture or country. The software project manager is required to be conscious of a summary of skills necessary at the right time.

Conclusion

It is generally well known that software projects suffer from failure to meet deadlines, cost overruns and leave customers disillusioned. Substantial effort has been made in the area of hard skills comprising processes, tools and techniques. Project management methodologies are being developed and improved, but still a large number of projects do not live up to expectations.

Apart from hard skills, research in the area of soft skills is very promising. Belzer (2004) mentioned that a greater piece of the puzzle for the successful delivery of projects, and in particular software projects, is that of soft skills. Companies like Mastek, Polaris and Sun Microsystems have incorporated such skills into their training agenda (Arora, 2003). A range of soft skills attributes required by a software project manager has been identified. The emotional intelligence framework proposed by Goleman (1998a) provides a number of soft skills dimensions that can improve software project success.

Alternatively, a table of soft skills, such as table 3 may be prepared to emphasize the skills required at various stages of the project in execution. This table could require further research, as it might not include the same soft skills for every organization, culture or regional setting. Furthermore, an empirical study as the one carried out at the University of Arkansas may be used as a starting point towards the accommodation of soft skills in software project management.

The soft skills required at different phases of a software project development life cycle need to be identified and the experienced software project manager has to be aware about the right time to make use of such skills. The soft skills, deemed important for software projects, have to be carefully selected and taken into consideration through the life cycle of a project. A simple framework has been proposed for an activation of the necessary soft skills during various phases of a software project. Accommodating soft skills in software project management can overcome some shortcomings in existing software project management methodologies. Therefore, greater research efforts should be expended to accommodate soft skills in software project management to improve success of the resulting projects. The outcome of these efforts may ensure greater customers satisfaction, motivate software project teams and exceed stakeholders' expectations.

References

Arora, S. (2003). Soft skills training: Outsourced vs. in-house. Retrieved February 03, 2005, from http://www.expressitpeople.com/20040426/cover.shtml

Barry, M. L. & Du Plessis, Y. (2004). An investigation into emotional intelligence as an important attribute for project managers. *Proceedings of the 2004 PMSA Global Knowledge Conference*, Johannesburg, South Africa, 320-329.

Belzer, K. (2004). Project management: Still more art than science. Retrieved August 19, 2004, from http://www.pmforum.org/library/papers/BusinessSuccess.htm

Bruce A. & Langdon, K. (2000): *Project management*. New York: Dorling Kindersley.

Bupa, S. H. (2005). Why do so many major IT projects fail? *Computer fraud & security*, 15-17.

Chapman, A. (2001). Stress management. Retrieved November 24, 2004, from http://www.businessballs.com/stressmanagement.htm

Cooke, R. (2004). Social psychology in project management. Retrieved August 23, 2004, from http://www.pmforum.org/library/papers/pmpsych1.htm

Dobbins, R. & Pettman, B. O. (1997). Self-development: the nine basic skills for business success. *Journal of management development, 16*(8), 521.

Dorsey, P. (2000). *Top 10 reasons why systems projects fail*. Retrieved February 10, 2005, from http://www.dulcian.com/papers

Farago, J. & Skyrme, D. (1995). The learning organization. Retrieved November 24, 2004, from http://www.skyrme.com/insights/3lrnorg.htm

Goleman, D. (1997). *Emotional Intelligence. Why it can matter more than IQ*. New York: Bantam Books.

Goleman, D. (1998a). *Working with emotional intelligence*. New York: Bantam Books.

Goleman, D. (1998b). What makes a leader? *Harvard Business Review, 76*(6), 93.

Gray, C. F. & Larson E. W. (2000). *Project Management: The managerial process*. Singapore: McGraw-Hill.

Heikkila, P. (2002). Skills survey 2002. You're in IT-Should you do an MBA? Retrieved February 03, 2005, from http://comment.silicon.com/0,39024711,11032240,00.htm

Hughes, B. & Cotterell, M. (2002). *Software project management* (3rd ed.). London, UK: McGraw-Hill.

James, D. (1999). *Managing people in organisations*. Financial World Publishing.

Kirsch, L. J. (2004). *Software project management: An integrated perspective for an emerging paradigm*. Retrieved August 19, 2004, from http://www.pinnaflex.com/pdf/framing/CH15.pdf

Lussier, R.N. (1990). *Human relations in organisations*. Homewood, IL: Irwin.

Moreira, P. (2004). *Soft skills for IT professionals*. Retrieved August 26, 2004, from http://www.certmag.com/issues/jan01/dept_techcareers.cfm

Newstrom, J. W. & Davis, K. (2001). *Organisational behavior: Human behavior at work*. New York: McGraw-Hill.

Oxford Advanced Learner's Dictionary. (1999).

Phillips, J. (2002). *IT project management*. New Delhi, India: Tata McGraw-Hill.

PMI. (2000). *A guide to project management body of knowledge*. PMBOK® Guide. 2000 Edition.

Proehl, R. A. (2001). *Organisational change in human services*. UK: Sage Publication.

Santrock, J. W. (2000). *Psychology* (6th ed.). Congress Lib.

Sonnekus, R. & Labuschagne, L. (2004). Establishing the relationship between IT project management maturity and IT project success in a South African context. *Proceedings of the 2004 PMSA Global Knowledge Conference*, Johannesburg, South Africa, 183-192.

Sukhoo, A., Barnard, A., Eloff, M.M., & Van der Poll, J.A. (2004a). Towards a framework for evaluating strengths and weaknesses of software projects. *Proceedings of the 2004 PMSA Global Knowledge Conference*, Johannesburg, South Africa, 193-199.

Sukhoo, A., Barnard, A., Eloff, M.M., & Van der Poll, J.A. (2004b). A survey of project management tools, techniques and methodologies used in Mauritius: The current status. *Proceedings of the 2004 PMSA Global Knowledge Conference*, Johannesburg, South Africa, 320-329.

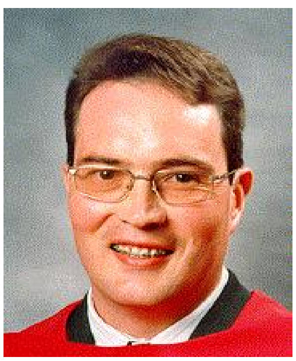
The Standish Group (2001). *Extreme CHAOS*. Retrieved August 25, 2004, from http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf

Fowdar, R. R. R., Peerally, J. A., Boolaky, M., Baichoo, R., Lai Wah, D., Gokhool, D., Vencatachellum, I., & Seebaluck, A. (2004). Organisation and management. Course manual. University of Mauritius.

Biography

Aneerav Sukhoo, is a project manager working for the Ministry of Information Technology & Telecommunications in Mauritius. He has 15 years of experience, out of which 8 years in IT project management, in dealing with implementation of IT projects in the Public Service.

He has a Honours Bachelor degree in computer Science from the University of South Africa and a Master of Information Technology degree from Charles Sturt University. He is presently preparing a PhD thesis at the University of South Africa for the development of a software project management methodology.



Andries Barnard, associate professor in the Department of Computer Science and Information Systems, holds a PhD (Computer Science). He teaches undergraduate courses in automata theory and formal languages and project management, as well as postgraduate courses in project management and research methodology. His research interests include computer ethics and graph grammar languages.

Mariki Eloff received a PhD Computer Science degree from the Rand Afrikaans University (RAU), South Africa. She has presented research papers at international and national conferences focusing on information security. She joined Unisa as an associate professor in the School of Computing during October 2002. She has assisted in the organization and management of international conferences in information security and has been part of the ISSA Conferences since its inception in 2000.

During 2001 she participated in information security management research projects under the joint leadership of RAU and the South African Bureau of Standards (SABS). She has also been engaged in the development of various information security-training modules for industry.

John A. Van der Poll is an associate professor and acting head of the Department of Systems Organisation in the School of Computing at the University of South Africa (UNISA). He holds a PhD in Computer Science and teaches an undergraduate course in operating systems, as well as postgraduate courses in compiler construction and formal program verification. His research interests are in formal specification languages and automated reasoning.

Mahendrenath Motah is senior lecturer at the University of Technology since 2000. He has extensive experience in the fields of Management, Administration, and Training at both national and international level. He has participated in various workshops, seminars and conferences at national and international level. He contributes to the local press and other media through articles and live programs. He has Master's degrees in Psychology and Ethnology, and a PhD in Arts and Human Sciences. He is also the Chairperson of the Regional Center for Urgent Anthropological and Ethnological Research (UNESCO, Vienna).

Interdisciplinary Journal of Information, Knowledge, and Management Volume 2, 2007

An Evolutionary Software Project Management Maturity Model for Mauritius

Aneerav Sukhoo, Andries Barnard, Mariki M. Eloff, and John A. Van der Poll
University of South Africa, Pretoria, South Africa

aneeravsukhoo@yahoo.com barnaa@unisa.ac.za
eloffmm@unisa.ac.za vdpolja@unisa.ac.za

Abstract

Software project management is a relatively recent discipline that emerged during the second half of the 20th century (Kwak, 2003). Many of the software project management methodologies available today were developed in Western/European countries and research showed that there was a need to formalise a software project management framework for developing countries, in particular Africa (Muriithi & Crawford, 2003).

Based on surveys and discussions with software professionals, a methodology for software project management is being proposed. The methodology is based on a maturity model as Mauritius is faced with a shortage in skilled professionals. So far, few organisations in Mauritius have been found to be using software project management methodologies developed in Western/European countries.

Most maturity models, for example Capability Maturity Model Integration (CMMI) and Kerzner's maturity model, have five maturity levels. The trend is towards the development of maturity models that have fewer maturity levels. For example, the Organisational Project Management Maturity Model (OPM3) and Prince 2 Maturity Model have been developed with three maturity levels.

Similarly, the proposed Evolutionary Software Project Management Maturity Model (ESPM^3) has three maturity levels and a continuous process improvement group of key process areas (KPAs). ISO 9001:2000 is chosen as the quality management system and each KPA is mapped onto the plan-do-check-act (PDCA) cycle. The model is conceptually represented as a conical structure to better display the evolutionary development of KPAs. KPAs, which are focus areas to be considered for project management, are developed until they attain maturity at a certain level. As organisations have to be responsive to their environments, these KPAs may require further changes even after they have reached maturity. A continuous process group of KPAs helps in adapting to changes in environment.

In order to test the proposed methodology, one case study has been included. The application of the methodology to the project could not be discussed in detail in this paper. The development of KPAs in an organisation takes time and the case study was only a snapshot of the application of the methodology.

Editor: Kathy Lynch

Keywords: software project management, software development, maturity model, evolutionary process, project management methodology.

Introduction

There is ample evidence that the inherent principles of project management have been practiced already for thousands of years (Kwak, 2003). The results are that many human achievements, for example the pyramids of Egypt and the Great Wall of China, are recognised as wonders of the present-day world. It is unfortunate that the Manhattan project (Kwak, 2003), although considered as a successful human undertaking (the development of the atomic bomb) at a later stage caused the loss of lives on a large scale.

Apart from the construction and engineering areas, applied principles of project management made inroads into virtually all avenues of work by encompassing sectors such as the military, social and community development, medicine, agriculture and education to name just a few (Kwak, 2003). Today, the impact of project management is most profound in the area of Information Technology (Gray & Larson, 2000) where new hardware and software products are constantly flooding the world market. Increased pressure to reduce cost and delivery time in a highly global and competitive environment has given due credit to project management principles, techniques and tools. Software project management, in particular, is an area of research with a view to achieve higher levels of quality and to improve both cost and schedule estimates. This is evident from frequent new releases of project management methodologies like the Project Management Body of Knowledge® (PMBOK® Guide, 2004), PRojects IN Controlled Environment (PRINCE) (Central Computer and Telecommunications Agency [CCTA], 2002) and the Capability Maturity Model Integration (CMMI Product Team, 2002). These methodologies were studied prior to the development of a suitable methodology for Mauritius. A software project management methodology is being proposed since it has been noted that a software project has certain specificities that are different from traditional engineering projects. Hughes and Cotterell (2006) argued in favour of the specificities of managing software projects. Some pertinent differences between software projects and traditional engineering projects were also elaborated on by Sukhoo, Barnard, Eloff and Van der Poll (2004b).

A plethora of project management methodologies is now available and many methodologies are subject to continuous improvement. Project management methodologies are generally generic in nature, in that they cut across various disciplines and are used in many countries. Some studies, for example Muriithi and Crawford (2003) and Stuckenbruck and Zomorrodian (1987), have revealed that such methodologies are not necessarily universally applicable because factors like economic rationalities are too often assumed and legal, political as well as cultural and religious variations occur across nations and cultures. Hofstede's dimensions as applied to African countries (Muriithi & Crawford, 2003), Kuwait (Aladwani, 2002) as well as Mexico, India and Russia (Rao, 2004) have been discussed in terms of high power distance and high uncertainty avoidance as compared to Western/European countries. Open discussions are not favoured and low tolerance for ambiguities is noted in developing countries. It is plausible to expect that these factors may negatively influence the successful outcome of software projects, particularly in developing economies, as these factors are seldom taken into account by a particular methodology. By and large, developing countries constantly face problems of insufficient skilled staff, funds, and political and social incentives (Muriithi & Crawford, 2003). Mauritius, for example, is facing such problems at a crucial moment in its history as the Government has embarked on a vision to develop it into a "Cyber Island" while at the same time allowing Information and Communication Technologies (ICT) to emerge as the fifth pillar of the economy besides sugar, tourism, textile and the financial services (Eid, 2002).

In particular, some of the problems affecting the Mauritian IT sector are:

- lack of skilled human resources (BPO Secretariat, 2005)
- high labour mobility (BPO Secretariat, 2005)
- lack of training (BPO Secretariat, 2005)
- ageing population
- unstable economic situation (Muriithi & Crawford, 2003)
- methodologies of Western/European origins are not completely suitable and are too expensive to adopt for most organizations (Muriithi & Crawford, 2003)
- lack of management commitment
- lack of soft skills in management of projects
- organizations are characterised by tall hierarchical structures (Muriithi & Crawford, 2003)
- existing methodologies are considered to be too bureaucratic.

This paper aims to address some of the problems faced by Mauritius through the application of an evolutionary software project management maturity model (ESPM$^{3)}$ to the management of software projects. This methodology, being a streamlined one and aiming at simplifying software project management, may also be extended to other developing countries given that the problems identified are common and confirmed according to the research conducted by Muriithi and Crawford (2003). It is anticipated that this ESPM^3 will assist organizations by adopting a progressive implementation of various processes.

The layout of this paper is as follows: we present a discussion on the unsuitability of two popular project management methodologies namely PMBOK® and CMMI in a developing country context, particularly for Mauritius. The results of a survey/interview focusing at gathering information to be used in a software project management methodology for developing countries are discussed in the next section. Then, we introduce an evolutionary software project management maturity model (ESPM^3) and show how the methodology addresses some of the problems faced by Mauritius as a developing economy in particular. We conclude with an analysis and some pointers to future work.

Unsuitability of PMBOK® and CMMI for Mauritius

We speculate on the unsuitability of PMBOK® and CMMI for Mauritius. The PMBOK® Guide does mention that a project team should consider a project in the context of political, economic, demographic and cultural norms, but it does not explain how or when to handle them. The PMBOK® Guide, furthermore, increased the number of pages from around 200 to around 400 since the previous release. The number of processes involved also increased from 39 to 44, thereby adopting a rather complex systems view of project management in general, and software project management in particular (Wideman, 2005). Specifically, small projects can be affected by the number of processes involved. The project team may, therefore, be overloaded by the number of process areas. In Mauritius, the lack of skilled software developers within organizations coupled with high labour mobility has a marked influence on the applicability of PMBOK®.

CMMI provides progressive improvements of organizational maturity by achieving maturity at a pre-defined level before advancing to a higher level. Hence, the project team focuses on a restricted set of process areas to reach a maturity level. However, CMMI does not assist in the management of influences brought about by social, economic or environmental pressures as well as human resource problems. In addition, the methodology incorporates a sheer volume of documentation of over 600 pages that increases the learning duration. Given that the interaction with human beings is important for a successful software development process, CMMI alone cannot address the human resource management problem. In Mauritius, not many companies can afford to adopt CMMI given the high initial cost of training required. Thus far, only one Mauritian organization is using this methodology (State Informatics Ltd Website, 2006).

Although the sheer volume of documentation of PMBOK® and CMMI was meant to improve the methodologies, most Mauritian companies have not been able to take advantage of them as the number of skilled personnel and funds for training are not sufficient.

According to the survey on project management tools, techniques and methodologies used in Mauritius (Sukhoo, Barnard, Eloff, & Van der Poll, 2004a), it was observed that there was resistance to use methodologies of Western/European origin. This is also in line with the research conducted by Muriithi and Crawford (2003).

In the next section we discuss a survey conducted in Mauritius to gather information to assist in the development of a software project management model followed by a description of the proposed model referred to as the Evolutionary Software Project Management Maturity Model ($ESPM^3$). It is presented as a simple and flexible methodology aimed at incorporating minimal bureaucratic procedures.

Survey/Interview Conducted in Mauritius

Two surveys were conducted , one on project management tools, techniques and methodologies used in Mauritius (Sukhoo et al., 2004a) and the other on software project maturity in Mauritius (Sukhoo, Barnard, Eloff, & Van der Poll, 2005b). In addition, a recent combined survey/interview (questionnaire is found in the Appendix) was conducted with 11 pertinent software development organizations in Mauritius during the period July to August 2006 with the aim at gathering information on KPAs, that is focus areas to be considered during management of software projects. All the respondents agreed on the applicability of a maturity model with less than 5 maturity levels for Mauritius. It is argued that a maturity model may hide certain deficiencies, but an organization is free to consider all KPAs for the proposed maturity model. The different levels help an organization to concentrate on certain KPAs at a level until it attains maturity. Then, the next level is considered. Organisations are able to identify their strengths and weaknesses and consider their areas of improvement. In the case of CMMI, which has 5 maturity levels, not many organizations in the world have reached the highest level. Therefore, the proposed model is meant to allow all Mauritian software development organizations to attain the highest level in the shortest time possible in order to develop the Information and Communication Technologies sector. Even the recently developed PRINCE 2 maturity model has adopted less than 5 levels. Three maturity levels (level 1 to 3) together with a set of continuous improvement key process areas were identified to be the most appropriate number of maturity levels. Key Process Areas (KPAs) to achieve a particular maturity level were also identified and eventually classified in different maturity levels.

A list of KPAs was provided in the questionnaire and most respondents agreed to its applicability. In this context, the most applicable KPAs identified in the Mauritian context were found to be the following:

- Time management
- Cost management
- Quality management
- Human Resource management
- Risk management
- Soft skills management
- Contract management
- Change management
- Software specific focus

- Integration management
- Environmental management

The respondents' classification of the KPAs into levels was as per Table 1:

Table 1: Survey/Interview results for classification of KPAs

	Respondents classification of KPA's at the different levels			Total no of re-spondents	Percentage response
	Level 2	Level 3	Continu-ous proc-ess im-prove-ment		
Time management	8	0	3	11	100%
Cost management	5	2	3	10	91%
Quality management	7	1	3	11	100%
Human resource manage-ment	1	7	2	10	91%
Risk management	2	7	1	10	91%
Soft skills management	2	3	5	10	91%
Contract management	2	7	0	9	82%
Change management	1	3	7	11	100%
Software specific focus	3	1	6	10	91%
Integration management	0	1	10	11	100%
Environmental management	9 out of 11 mentioned that a KPA dealing with cultural, political and economic aspects was important.				82%

It was noted that some respondents did not provide any classification for some KPAs. It can, however, be noted that above 80% of responses were received for classification of KPAs. For Time management, 8 respondents mentioned that the KPA had to be considered at level 2, no respondent mentioned that the KPA had to be at level 3 and 3 respondents mentioned that the KPA had to be classified into the continuous process improvement group. Therefore, Time management was considered as a maturity level 2 KPA (that is, the level at which the majority of respondents agreed that the KPA should be). The continuous process improvement group, however, comprises KPAs that need to be taken into consideration at all maturity levels (at least informally at level 1 as well).

According to the above-mentioned combined survey/interview as well as the 2004 (Sukhoo et al., 2005b) survey, the key process areas identified were categorised into two levels (2 and 3) as mentioned previously. Other KPAs were found to be important at all levels and they act as factors, which triggered improvements for KPAs at both levels 2 and 3. For instance, Change management can have an impact on the Time management KPA in case political pressure is exerted to motivate project delivery before the scheduled time.

The different KPAs are therefore grouped into maturity levels as shown in Table 2.

Table 2: KPAs grouped in maturity levels

KPA	**Maturity level**
Time management	2
Cost management	2
Quality management	2
Human resource management	3
Risk management	3
Soft skills management	Continuous process improvement group
Contract management	3
Change management	Continuous process improvement group
Software specific focus	Continuous process improvement group
Integration management	Continuous process improvement group
Environmental management	Continuous process improvement group

Most of the above KPAs were also considered important success factors in a previous study conducted in Mauritius (Sukhoo et al., 2005b).

Furthermore, each KPA is mapped onto four process groups as per ISO 9001:2000 (Mauritian Standard, 2001) requirements as follows:

(i) *Plan*: Establish the objectives and processes necessary to deliver results in accordance with customer requirements and organizational policies.

(ii) *Do*: Implement the processes.

(iii) *Check*: Monitor and measure processes and products against policies, objectives and requirements and report the results.

(iv) *Act*: Take actions to continually improve process performance.

ISO 9001:2000 was adopted as the quality management standard for our proposed ESPM^3 because many organizations in Mauritius are already following this standard (Management Audit Bureau Website, 2005).

Following the identification and grouping of the KPAs into different maturity levels, a conical structure is used to represent the maturity levels comprising these KPAs in the next section.

The Evolutionary Software Project Management Maturity Model (ESPM^3)

The conceptual representation of the proposed ESPM^3 adopts a conical structure with three defined levels (maturity levels 1 to 3), depicted in Figure 1. The conical structure has been chosen as it better displays the evolutionary development of the KPAs.

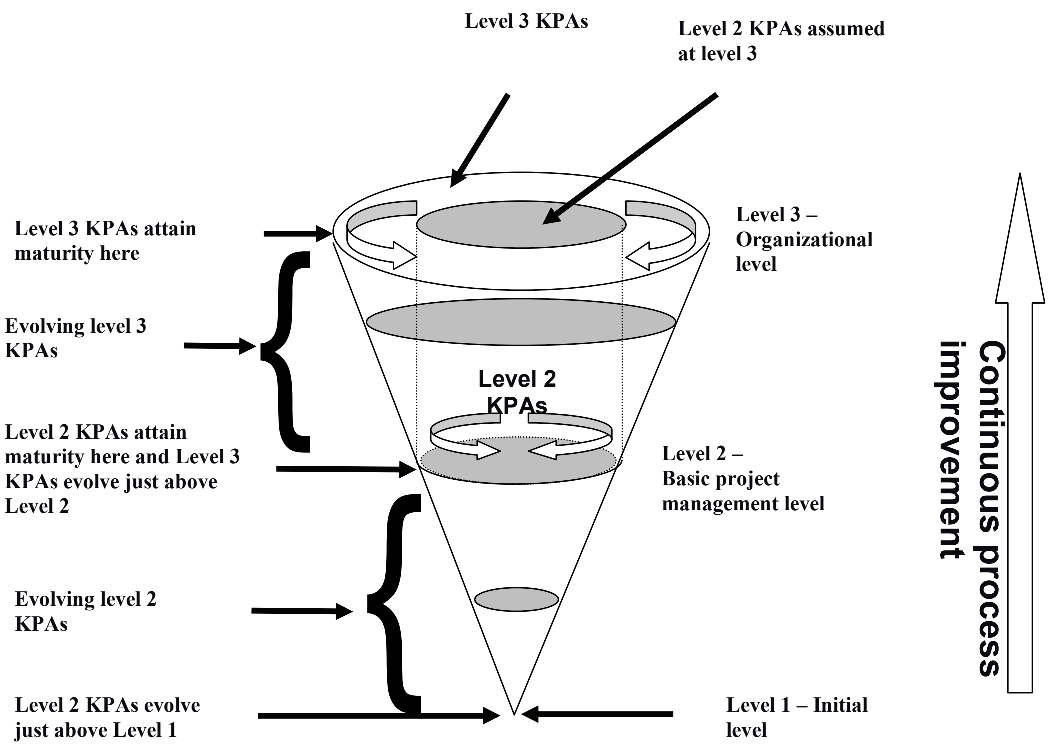


Figure 1: Conceptual Representation of ESPMMM

Maturity levels 2 and 3 each consist of a number of KPAs, which are focus areas in order to achieve the respective maturity level. Maturity level 1 is the initial level of our model and has no KPAs associated with it since software development at this level is carried out in an ad-hoc manner. In other models this level is often considered as the "chaotic" level. The KPAs at levels 2 and 3 are represented conceptually as sectors on the horizontal section of the cone (Figure 7). KPAs at a specified maturity level are assumed at higher levels. For example at maturity level 3, a cross section of the cone reveals the KPAs at level 2 as well as the new KPAs at level 3. Note, however, that the pictorial representation of an area of a sector is not representative of the amount of effort required to implement a specific KPA.

Specific to problems identified in some Mauritian software development companies, a range of key process areas was identified to be implemented at the various maturity levels introduced above. Specific shortcomings pertaining to some or all of these KPAs were identified during surveys conducted in Mauritius in 2003 (Sukhoo et al., 2004a) and 2004 (Sukhoo et al., 2005b). In 2003, 50% of Mauritian software development companies mentioned that more than 50% of their projects failed to meet deadlines, 30% of companies mentioned that more than 10% of their projects suffered from budget overruns and 10% of companies pointed out that more than 75% of their projects did not satisfy the users with regard to quality. Risk management, inappropriate project team, poor planning, lack of skilled professionals, mobility of labour, communication problems among team members as well as with clients, and the lack of an appropriate methodology were among several problems that were encountered. Furthermore, the Risk management KPA was not adequately addressed, since most companies surveyed at that point in time remained at maturity level 1.

Representation of KPAs throughout the Evolutionary Process

There are some KPAs that are not tied to a specific maturity level but need to be considered throughout the evolutionary process. This continuous improvement group of processes spans over all three levels. The continuous improvement process group, M_c, is defined generally as the union of a number of KPAs.

$$M_c = \bigcup_{k=1}^{p} KPA_{ck} \qquad (1)$$

where p is the total number of KPAs in the continuous process improvement group. A KPA in M_c is identified by adding the subscript "c" to the specific KPA, for example KPA_{c1} is the first KPA in M_c.

For ESPM[3] we identify 5 KPAs that need to be considered at all maturity levels. This selection is based on, amongst other things, respondent classifications of the KPAs as per the questionnaire (refer to Table 1). These KPAs are:

- Soft skills management (KPA_{c1})
- Change management (KPA_{c2})
- Software specific focus (KPA_{c3})
- Integration management (KPA_{c4})
- Environmental management (KPA_{c5}).

The above KPAs are expected to influence all other KPAs and induce continuous improvement. They are represented by the arrow on the right in Figure 2.

The Environmental management KPA is concerned with specific social, cultural, political and economic factors for Mauritius. These factors were found to affect software project management. For example, a project deemed to offer certain political benefits need to be given high priority and must be managed with extreme care. In managing the environmental factors, the following inputs are transformed into outputs:

- List of environmental factors to be addressed (for example political, social, cultural and economic benefits so that such projects are given priority). An example of such a project is treated in the case study.
- List of ways to deal with the factors (for example, assigning skilled human resources and technology)
- Historical data.

The transformation yields the following outputs:

- Satisfied team members and stakeholders
- Safe software development environment (for example, other projects must not fail due to political, social and economic priorities)
- Lessons learnt.

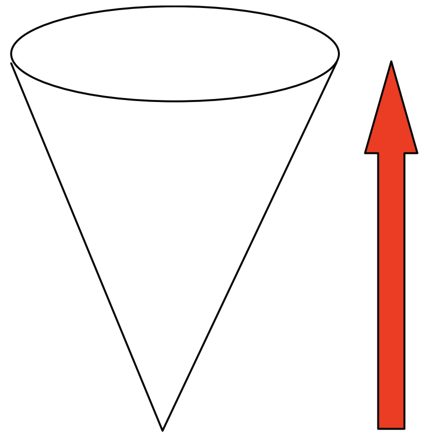


Figure 2: Continuous process improvement

The continuous improvement process group for ESPM[3], M_c, is the union of Soft skills management (KPA_{c1}), Change management (KPA_{c2}), Software specific focus (KPA_{c3}), Integration management (KPA_{c4}), and Environmental management (KPA_{c5}).

Instantiating p in formula (1) above for the number of KPAs at this level, the continuous improvement process group for ESPM[3] is given by:

$$M_c = \bigcup_{k=1}^{5} KPA_{ck} \qquad (2)$$

Maturity Level 1

Maturity level 1 is the *initial level* in the proposed model, as shown in Figure 3.

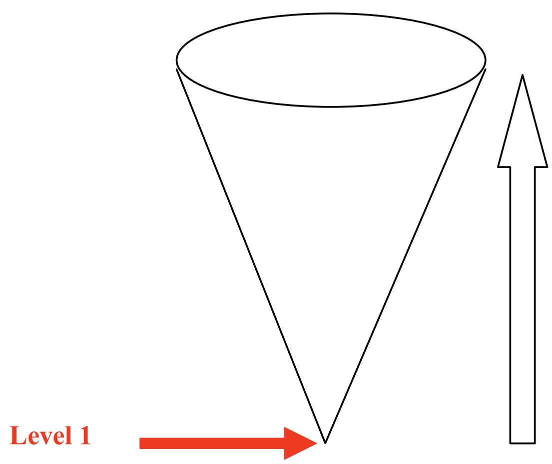


Figure 3: Maturity level 1

At this level, no KPAs are defined. The software development process is carried out in an ad-hoc (chaotic) manner similar to other maturity models and project success depends on the project team. A change in the project team while the project is in progress may disrupt its execution as the skills and processes used are not consistent among project teams. The operation of an organization at level 1 involves uncalculated risks and progress towards level 2 is necessary. Maturity level 1, namely M_1, contains no KPAs, that is $M_1 = \emptyset$ or it is the null set.

Maturity Level 2

This level provides a basic project management focus, which is important for the software development process, since it forms the foundation for level 3. It is referred to as the basic project management level. Figure 4 depicts level 2 as a horizontal section of the cone.

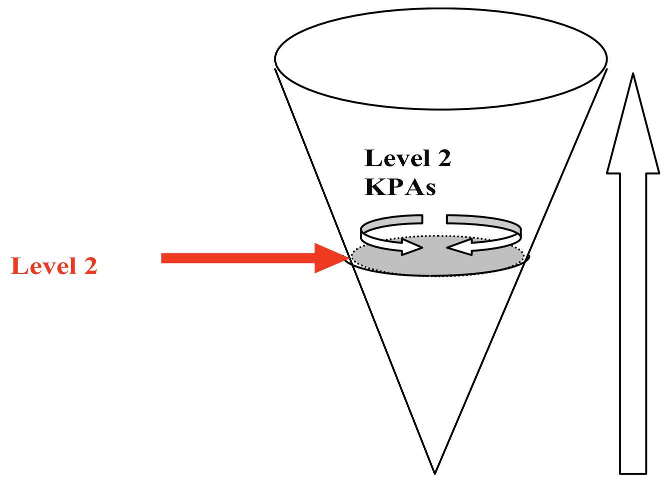


Figure 4: Maturity level 2

Maturity level 2, namely M_2, is the union of all KPAs at level 1 (M_1), all KPAs at the current level and the continuous improvement process group of KPAs (M_c), that is

$$M_2 = M_1 \cup \bigcup_{j=1}^{n} KPA_{2j} \cup M_c$$

M_1 and M_c are given by (3) and (1) respectively and n is the total number of KPAs at level 2.

Figure 5 shows graphically how the KPAs (say n KPAs) at level 2 are represented as a view from the top of the cone. In addition the continuous improvement group of KPAs, M_c, influences the level 2 KPAs.

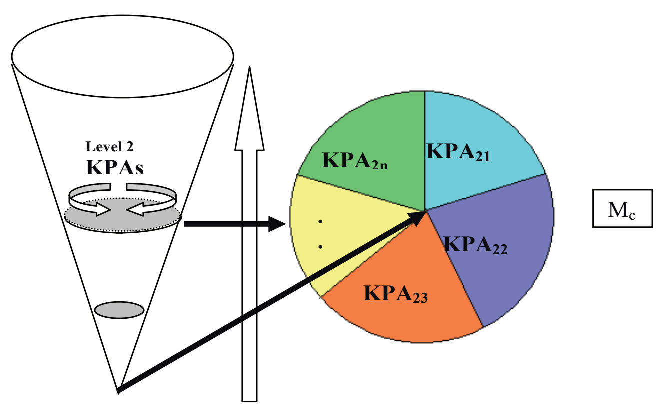


Figure 5: Conceptual Representation of n KPAs at level 1 and 2

Assuming a maturity level with n KPAs, each one is represented as a sector by viewing the appropriate horizontal section of the cone. At a specified maturity level, its associated KPAs are considered to have evolved fully and do not require further development at any of the higher levels except for continuous process improvement where improvement is required for the process to adapt to the dynamic environment in which organizations evolve.

A KPA at level 2 is indicated as KPA_{2i}, for example, KPA_{23} is the third KPA at maturity level 2. Only the essential KPAs for the basic project management level are selected based, among other things, on the respondent classification in the survey conducted in order to encourage organizations to progress rapidly to level 2 and ensure that an acceptable number of projects are delivered on time, within budget and according to a specified quality standard. In this context for ESPM[3], and based on respondent classifications (see Table 1), three KPAs are chosen to achieve this level, and they are:

- Time management: to ensure timely delivery of projects (KPA_{21})
- Cost management: to deliver projects within budget (KPA_{22})
- Quality management: to deliver projects according to ISO 9001:2000 standards (KPA_{23}) and to meet customer requirements.

Substituting for n in formula (3) and for $M_1 = \emptyset$, M_2 for ESPM[3] is defined as

$$M_2 = \bigcup_{j=1}^{3} KPA_{2j} \cup M_c \qquad (4)$$

Maturity Level 3

This level provides an organizational focus. Maturity level 3, M_3, is the union of all KPAs at level 3 and level 2 (M_2), where M_2 already includes the continuous improvement process group of KPAs (M_c). This level is shown in Figure 6.

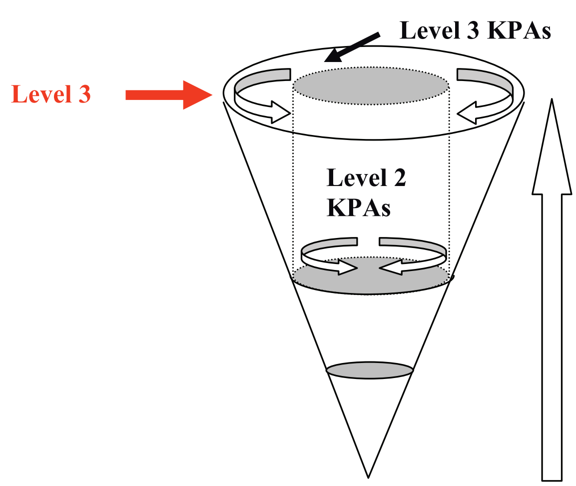


Figure 6: Maturity level 3

Therefore,

$$M_3 = \bigcup_{i=1}^{m} KPA_{3i} \cup M_2 \qquad (5)$$

where m is the total number of KPAs at level 3 (the KPA_3's) and M_2 is given by (4).

In general, for ESPM[3], containing any number of maturity levels, the i[th] maturity level, that is M_i, is defined as

$$M_i = \bigcup_{j=2}^{i-1} \{ M_j - M_c \} \cup \{ \bigcup_{k=1}^{m} KPA_{ik} \} \cup M_c$$

where i, j, k and m are natural numbers, M_c is the continuous improvement process group of KPAs and KPA_{ix} is the x[th] process area associated within the i[th] level.

KPAs for level 3 are normally developed once level 2 has been reached and the organization concerned intends to reach level 3. In this model specific to the Mauritius context and guided by respondent classification summarised in Table 1, the following KPAs are developed at level 3:

- Human Resource management (KPA_{31})
- Risk management (KPA_{32})
- Contract management (KPA_{33})

Instantiation of the KPAs at level 1, 2 and 3 leads to the arrangement of the KPAs depicted graphically in Figure 7.

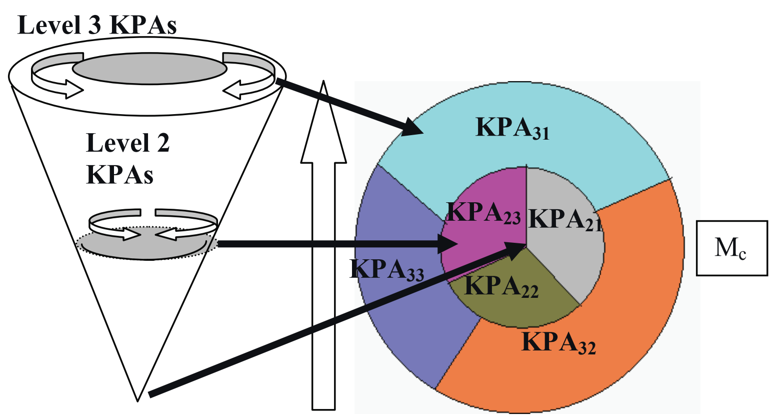


Figure 7: Instantiated representation of KPAs at level 1,2 and 3

Substituting for m in formula (5) defines M_3 for ESPM[3] as:

$$M_3 = \bigcup_{i=1}^{3} KPA_{3i} \cup M_2 \qquad (6)$$

Here KPA_{21}, KPA_{22} and KPA_{23} are the key process areas that attained maturity at level 2 (M_2), while KPA_{31}, KPA_{32} and KPA_{33} are the key process areas that attained maturity at level 3. In general, therefore, KPAs at level i evolve after efforts are made to reach level i from level i-1. Thus, any KPA_{in} for valid values of i and n, matures at level i.

An organization is, however, not prevented from developing and using a KPA at a higher level, for example an organization at level 2 may include a KPA of level 3 should this KPA be considered important. Note that the organization cannot claim to be at the higher level if this happens, but it can be referred to as an extended level. For example, an organization at level 2 which is using a KPA of level 3 will be referred to as level 2-extended.

Mapping of KPAs to the PDCA cycle

As mentioned before, each KPA is mapped to the PDCA cycle. This is shown in Figure 8.

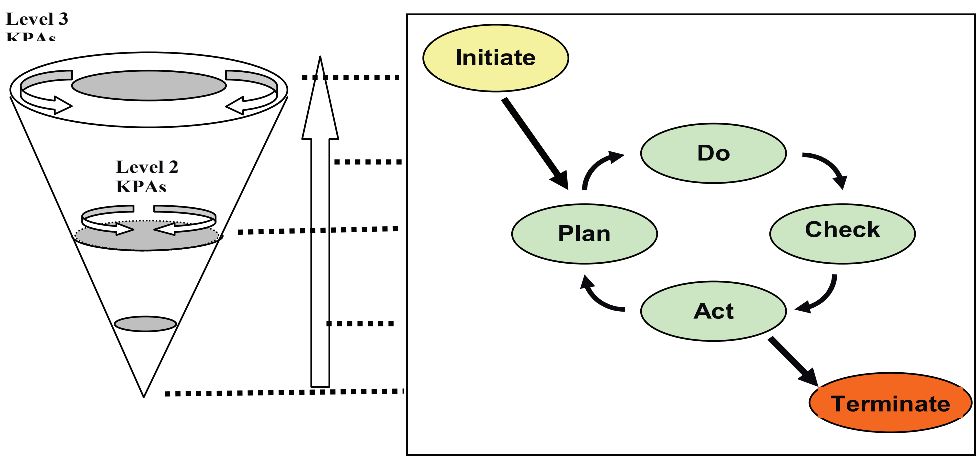


Figure 8: Mapping of KPAs to the PDCA cycle

Suitability of ESPM³ for Mauritius

The proposed ESPM³ addresses the problem of human resources, soft skills, contract management as well as software specificities. It also takes into consideration the political, social, economic and cultural aspects associated with software project management. In the Mauritian context, it allows software development organizations to focus on a few KPAs initially (for example 3 KPAs till maturity level 2 is reached) and subsequently adding more KPAs to progress to the next higher level (that is level 3). An organization at a lower level may also include a KPA of a higher level and enable it to operate at the extended lower level. The continuous process improvement group comprises those KPAs that affect KPAs at levels 2 and 3. For example, Integration management binds all KPAs in a coherent manner and the Environmental management KPA makes use of the political factor, when relevant, to influence Time management. The Software specific focused KPA influences the way project management is carried out for software development, example prototyping is considered as an important aspect in software development. KPAs are mapped to the Plan-Do-Check-Act process group as per ISO 9001:2000 requirement (Mauritian Standard, 2001). PMBOK® and CMMI include numerous KPAs (in the case of PMBOK, the KPAs called knowledge areas are further subdivided into 44 processes) to be considered and therefore the implementation of such methodologies is time consuming and renders the software development process complex in the Mauritian context. Thus, the $ESPM^3$ is an attempt to reduce much of the bureaucracy associated with other models.

The evolutionary process is better shown in $ESPM^3$ (for example in figure 1) than in CMMI.

The next section shows the application of $ESPM^3$ to a project in the Mauritian context.

Case Study

A case study is included below to test the applicability of the $ESPM^3$. It should be noted that each KPA is applied using the Plan-Do-Check-Act cycle that is a requirement of ISO. Each KPA is planned, executed, monitored and corrective actions are taken. However, due to time and space constraints, these specific details are omitted from this paper.

Allowance Computation System

This project comprises the selection of candidates for payment of an allowance earmarked for the low-income population group of Mauritius. An allowance amounting to US$ 1.3 as proposed for needy beneficiaries of basic pensions (Basic Retirements Pension, Basic Widow Pension, Basic Invalidity Pension and Basic Orphan Pension). This project was chosen because of its political, social and economic attributes.

The implementation of the system had to be carried out at the earliest possible time during the financial year 2006/2007 given the political benefits to be derived and the relief it ought to provide to this segment of the population.

This project comprised tasks encompassing analysis, design, software development, implementation and training.

The company responsible for the development of the software and its implementation endeavoured to make use of CMMI in the past already and was, therefore, familiar with progression from one maturity level to the next. The $ESPM^3$ was tested for the development of the software. All the KPAs were applied.

The following estimates were initially made:

Time frame: The estimated time frame for completion of the project was 5 weeks

Proposed start date and end date: 11 August 2006 to 14 September 2006

Estimated no. of person days: 64 person days.

Results obtained from implementation of the proposed $ESPM^3$ showed encouraging results as described below:

- **Level 2 KPAs**
  - Time management:

  A Work Breakdown Structure (WBS) was applied to estimate the time to complete the project in terms of person-months. Historical data was used to estimate the time to complete similar tasks obtained from the WBS (time estimation by analogy).

  Human resources had to be made available given the political pressure exerted. The environmental KPA dealing with the political aspect was considered. This involved the planning process for highly skilled software developers to be deployed on the project. A contingency plan was also devised to redeploy developers from other projects.

The ESPM[3] supports Time management by rendering the various tasks along a timeline using a Gantt chart. The actual progress could be verified against the planned progress and corrective action was initiated as soon as a significant deviation was noted.

Given the high power distance, resulting from the organization having tall organizational and/or report structures, it was important for the project leader to report problems to the head of the user organization in order not to delay the project. An example of a problem that had to be solved in this way was the frequent change in requirements by users interacting with the software developers. In the software development organization, the project leader acted as a facilitator for the project in addition to shouldering the responsibilities of the project.

- Cost management:

Using the set of components produced from the WBS, the cost of the software was estimated by taking into consideration the total number of person-months calculated during Time management.

- Quality management:

An evolutionary prototype was developed and user feedback was gathered and taken into consideration. This approach was not always adopted by the organization previously.

- **Level 3 KPAs**

  - Human Resource management:

Given the importance of the project to be delivered on time, efforts had to be made to carefully select staff possessing the required skills and experience. Some staff also had to be transferred from other projects. It was, however, not possible to prevent some developers to simultaneously work on multiple projects. Therefore, some staff had to work for extra hours and payment of overtime could not be avoided.

  - Risk management:

The main elements of risks included unavailability of key staff during the project, non-compliance to the terms of the contract and misunderstandings between the client and developers. The risks were evaluated in terms of severity of impacts and prioritized accordingly.

The Project Leader himself carried out the task of Risk management by ensuring that necessary human resources were adequate. Non-compliance to the terms of the contract could result mainly in the payment of a penalty fee. Misunderstandings were cleared up promptly with the use of the evolutionary prototype. The main issues that required action were arrangements of data fields in terms of a logical flow on the screen.

It was noted that the risks were successfully dealt with by the Project Leader.

Uncertainty avoidance, that is the extent to which ambiguous situation or uncertainty is avoided, was high in order to prevent any schedule overrun. However, the adoption of such a stand by the software organization concerned was necessary, that is, it was not possible to use new tools and techniques with which the staff had little or no experience for this project as no delay could be entertained.

- Contract management:

  A standard contract was signed between the company and the client and the Project Leader ensured compliance to the terms mentioned. Such a contract was a normal procedure for the company.

  However, $ESPM^3$ showed a new approach to handle Contract management with the mapping of the KPA to the PDCA cycle.

- **Continuous Improvement Process group of KPAs**

  - Soft Skills management:

  According to the Project Leader, the use of soft skills (such as good communication skills, team building, stress management and conflict management) has always been practiced by him and excellent results were obtained. It was not difficult to have staff working extra hours on projects. Staff members were discouraged from taking leave for the duration of such a project owing to the commitment of the company towards the successful implementation of the project that was considered of importance to assist the needy segment of the population as well as the reputation of the company. The staff responded positively to the request of the Project Leader.

  However, according to the Project Leader, it is a known fact that Mauritian employees show a strong concern for family and social commitments. The environmental management KPA dealing with the social aspect was found to be important.

  - Change management:

  The major problem encountered was that users kept changing their requirements throughout the duration of the project. It was important to have top management involvement to prevent a schedule overrun. Given the tall organization structure of the user and software development organizations, it was necessary to deal directly with higher officers to clear misunderstandings in some cases. Senior organization officers were also encouraged to participate in evaluation of prototypes.

  - Software specific focus:

  Use of reusable code was encouraged by interaction between developers on various projects and this task was facilitated by the Project Leader. This contributed a lot to reduction in development time.

  - Integration management:

  Integration Management, being an important KPA could not be disregarded for the project. Its application ensured that the other KPAs were well coordinated in order to deliver the project successfully.

  - Environmental management:

  It was noted from the above explanation that this KPA influenced other KPAs like Time management, Human Resource management and Soft skills management.

The visibility of tasks was enhanced with simple tools and techniques like the use of WBS, Gantt charts and an evolutionary prototype (that is, providing a software specific focus). High priority was given to the project due to the political pressure exerted. Skilled developers had to be allocated to the project and a contingency plan had to be devised for deployment of additional staff. The project manager had to use soft skills and recognise the collectivist nature (in terms of extended family ties) of the staff in order to motivate them. Therefore, the company had to manage its human resources carefully so as not to delay other projects. Power distance, that is the extent to which power in an organisation was unequally distributed, was high and problems had to be resolved by the Project Leader by raising issues rapidly upward in the user organization and at the same time assuming responsibility thereof.

Soft skills management has an effect on Human Resource management. In addition to managing changes with respect to the project, Change management is also expected to induce and manage changes in all other KPAs. Software specific focused KPAs also affect other KPAs – example Time management may be enhanced with new software tools. The continuous improvement group of KPAs could only show positive effects over some period of time.

The advantage of $ESPM^3$ was that there was no need for formal, long duration training for the project leader to apply the methodology. One session of face-to-face discussion and guidance by phone were enough. Furthermore, all the KPAs were applied successfully.

It was possible for the Project Leader to use both $ESPM^3$ and CMMI concurrently. According to the Project Leader, $ESPM^3$ was simpler and did not involve bureaucratic procedures as in the case of CMMI.

Conclusions and Future Work

This paper briefly referred to some of the problems experienced by Mauritius as a developing country with respect to delivery of successful projects. Mauritius is trying to develop its ICT sector as a pillar of its economy with the result that software development projects are growing rapidly in number. Based on earlier research, it was shown that project management methodologies of European/Western origins are not universally applicable (Muriithi & Crawford, 2003). In addition, previous work carried out by Sukhoo et al. (2004a) and Sukhoo et al. (2005b) illustrated that this was also the case in Mauritius. The problems faced by Mauritius were considered and a case study for the applicability of $ESPM^3$ was reported on.

The $ESPM^3$ provided a simple and flexible evolutionary approach for managing software development projects by portraying the evolutionary development of KPAs and at the same time rendering conceptually an integrated management process. The software specific focused KPA laid special emphasis on software projects by considering the strengths and weaknesses of such projects (Sukhoo et al, 2004b). The soft skills KPA was included to emphasize the importance of human relationships in management of software projects (Sukhoo, Barnard, Eloff, & Van der Poll, 2005a). In addition, KPAs were considered with particular attention to economic, cultural, social and political aspects to attain successful delivery of software projects.

The $ESPM^3$ was tested on three projects and one such project was reported on by means of a case study scenario as part of this paper. As future work, this model may be applied and improved further for larger projects. In addition, the progressive evolution of KPAs from one level to another can be portrayed by using $ESPM^3$ over a longer duration.

References

Aladwani, A. M. (2002). IT Project uncertainty, planning and success: An empirical investigation from Kuwait. *Information Technology and People*. *15* (3), 210-226.

BPO Secretariat – Board of investment. (2005). *Survey on the ITES BPO sector in Mauritius*.

Central Computer and Telecommunications Agency [CCTA]. (2002). *Project management industry initiatives*. Retrieved December 12, 2002, from http://www.ogc.gov.uk/prince/progmgtbrief1997.pdf

CMMI Product Team (2002). Capability Maturity Model Integration. Retrieved January 3, 2006, from http://cmsu2.cmsu.edu/public/classes/sam/cis4655/cmmi.pdf

Eid, A. (2002). Cyber Island, Silicon Valley under the tropics. *Islander*. *32*, 66-67.

Gray, C. F. & Larson, E. W. (2000). *Project management: The managerial process*. Singapore: Mc Graw-Hill.

Hughes, B. & Cotterell, M. (2006). *Software project management*. (4th ed.). London, UK: McGraw-Hill.

Kwak, Y. H. (2003). Brief history of project management. Retrieved September 19, 2005 from http://home.gwu.edu/~kwak/PM_History.pdf

Management Audit Bureau Website (2005). *ISO 9001 Implementation*. Retrieved January 4, 2006 from http://www.gov.mu/portal/site/mabsite/menuitem.b4fa590d611ae5b397da581065c521ca/

Mauritian Standard (2001). *Quality management systems – Requirements*. Mauritius Standards Bureau.

Muriithi, N. & Crawford, L. (2003). Approaches to project management in Africa: Implications for international development projects. *International Journal of Project Management*, *21*(5), 309-319.

PMBOK Guide (2004). *Third edition excerpt*s. Retrieved January 3, 2006 from http://www.pmi.org/info/pp_pmbok2k_conf.asp

Rao, M.T. (2004). Key issues for global IT sourcing: Country and individual factors. *Information Systems Management*, *21*(3), 16-21.

State Informatics Ltd Website (2006). SIL's Quality Policy. Retrieved December 8, 2006 from http://sil.intnet.mu/Achievements/quality.htm

Stuckenbruck, L.C. & Zomorrodian, A. (1987). Project management: The promise for developing countries. *International Journal of Project Management*, *5*(3), 167-175.

Sukhoo, A., Barnard, A., Eloff, M.M. & Van der Poll, J.A. (2004a). A survey of project management tools, techniques and methodologies used in Mauritius: The current status. *Proceedings of the 2004 PMSA Global Knowledge Conference*, 320-329.

Sukhoo, A., Barnard, A., Eloff, M.M. & Van der Poll, J.A. (2004b). Towards a framework for evaluating strengths and weaknesses of software projects. *Proceedings of the 2004 PMSA Global Knowledge Conference,* 193-199.

Sukhoo, A.; Barnard, A.; Eloff, M.M. & Van der Poll, J.A. (2005a). Accommodating soft skills in software project management. *Journal of Issues in Informing Science and Information Technology, 2,* 691-704. Available at http://2005papers.iisit.org/I55f42Sukh.pdf

Sukhoo, A., Barnard, A., Eloff, M.M. & Van der Poll, J.A. (2005b). An assessment of software project maturity in Mauritius. *Journal of Issues in Informing Science and Information Technology, 2,* 671-690. Available at http://2005papers.iisit.org/I54f41Sukh.pdf

Wideman, R. M. (2005). PMBOK Guide (3rd ed.) – Is more really better? Retrieved January 3, 2006 from http://maxwideman.com/papers/pmbok3/pmbok3b.pdf

Appendix – Questionnaire

Interview Questionnaire on the use of Software Project Management Maturity Model	
1.	Is your organization currently making use of a Maturity Model (MM)? If yes, proceed to question 2 otherwise proceed to question 4.
2.	Which type of MM (general or software specific) is your organization using?
3.	What are the problems that you encounter with the MM used in your organization?
4	Are you in favour of a software project management maturity model in particular v/s a general project management maturity model? Explain why yes / no.

5.	Maturity models make use of key process areas or focus areas in order to determine the maturity level of an organization. Are the following KPAs (or focus areas) important and should it ideally be considered by a software project maturity model: • Time management • Budget management • Quality management • Human Resource management • Risk management • Soft skills management • Contract management • Change management • Software specific focus • Integration management
6.	Do you think that other KPAs are also important for software project management in Mauritius? If so, list other KPAs and briefly explain why you consider them to be important.
7.	What are the features that you believe to be important in these KPAs that need to addressed?
8.	Many maturity models in use (like Kerzner's maturity model, Capability Maturity Model (CMMI) and Microframe's self assessment tool) have five maturity levels. Do you think that a MM with less maturity levels will assist more companies in Mauritius to adopt it?
9.	How would you arrange the KPAs in question 5 above as well as any other KPAs identified in question 6 in different maturity levels taking into consideration the number of levels envisaged as per question 8?
10.	Do you think continuous process improvement is important while dealing with KPAs or do they have to remain as static areas of a MM.
11.	Mauritius has a shortage of skilled manpower in the field of ICT. In your opinion, what are the possible solutions?
12.	Some studies have revealed the influence of culture (based on Hofstede's dimension as well as dealing with cross cultural aspects), economic situation, political situation (projects that have political benefits are given priority, social conditions (low wages – tendency to supplement with part time employment, poverty, family ties). Do you believe that the introduction of one or more KPAs dealing with these issues will assist in the management of software projects in Mauritius? Why yes or no?

Biography

Aneerav Sukhoo works as Acting Deputy Director for the Central Informatics Bureau, Ministry of Information Technology & Telecommunications in Mauritius. He has 17 years of experience in the IT field, out of which 10 years in IT project management, in dealing with implementation of IT projects in the Public Service. He has a B.Sc Honours degree in Computer Science from the University of South Africa (Unisa) and a Master of Information Technology degree from Charles Sturt University. He is presently preparing a PhD thesis at the University of South Africa for the development of a software project management methodology.

Andries Barnard, Associate Professor in the School of Computing at Unisa, holds a PhD (Computer Science). He teaches undergraduate courses in automata theory and formal languages and roject management, as well as postgraduate courses in project management and research methodology. His research interests include computer ethics and graph grammar languages.

Mariki M. Eloff received a PhD Computer Science degree from the Rand Afrikaans University (RAU), South Africa. She has presented research papers at international and national conferences focusing on information security. She joined Unisa as an associate professor in the School of Computing at Unisa in October 2002. She has assisted in the organisation and management of international conferences in information security and has been part of the ISSA Conferences since its inception in 2000. Her research interests include computer security and project management.

John A. Van der Poll is a full professor in the School of Computing at Unisa. He holds a PhD in Computer Science and teaches an undergraduate course in operating systems, as well as postgraduate courses in compiler construction and formal program verification. His research interests are in formal specification techniques and automated reasoning.