

## Middlesex University Research Repository:

an open access repository of  
Middlesex University research

<http://eprints.mdx.ac.uk>

Sandhawalia, Birinder Singh, 2008.  
An empirical investigation of knowledge management support for  
software projects.  
Available from Middlesex University's Research Repository.

---

### Copyright:

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this thesis/research project are retained by the author and/or other copyright owners. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge. Any use of the thesis/research project for private study or research must be properly acknowledged with reference to the work's full bibliographic details.

This thesis/research project may not be reproduced in any format or medium, or extensive quotations taken from it, or its content changed in any way, without first obtaining permission in writing from the copyright holder(s).

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:  
[eprints@mdx.ac.uk](mailto:eprints@mdx.ac.uk)

The item will be removed from the repository while any claim is being investigated.

# **An Empirical Investigation of Knowledge Management Support for Software Projects**

Birinder Singh Sandhawalia

Supervisors:

Prof. Darren Dalcher

Prof. Colin Tully

Dr. Stylianos Hatzipanagos

Thesis submitted in partial fulfilment of the requirements for the degree of  
Doctor of Philosophy

School of Computing Science, Middlesex University

March 2008

## **Acknowledgements**

I have received encouragement, guidance and support from a number of people in my work. Some people deserve a special mention.

First, my supervisors, Prof Darren Dalcher, Prof Colin Tully and Dr Stylianos Hatzipanagos; thank you all for having faith in my ability to conduct this research. Darren, thank you for all the encouragement, insights, understanding, patience, long hours and effort while guiding me through the research; you have been a great supervisor. Colin, thank you for your wisdom, encouragement and guidance; it has been a privilege to work with you. Stylianos, thank you for your encouragement, time, understanding and advice; I have benefitted a lot from your feedback and support.

I would like to thank the members of the Software Forensics Centre and the National Centre for Project Management. Thank you all for your support and interest in this research.

Thank you to the staff of the School of Computing Science, with a special mention to Dr Elke Duncker-Gassen, Dr Claudia Kalay, Pia Wallington, Dr Robert Pleass, Dr Christian Huyck, Prof David Turner, Elli Georgiadou, Pirkko Harvey, Elaine English, Yvonne Kidd, Nicola Savill, Kerry Gaulton, Emma Warne and Gerald Osei-Kofi.

To all my friends and colleagues who had a genuine interest in my work, especially, Gurteg Sandhu, Dr Saurabh Bansal, Manpreet Gill and Antoinette Ahiable for their co-operation.

Finally, Drish, Simran and Iqroop, for their unstinting love, help and encouragement; and my parents, Preminder and Kushal, to whom words cannot express how much I owe.

Thank you all for your guidance and support.

# Table of Contents

<b>I. ABSTRACT.....</b>	<b>7</b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>10</b>
1.1 ORIGINALITY OF WORK .....	14
1.2 ORGANISATION OF THESIS .....	15
<b>PART I LITERATURE REVIEW.....</b>	<b>17</b>
<b>CHAPTER 2 UNDERSTANDING KNOWLEDGE AND ITS MANAGEMENT.....</b>	<b>18</b>
2.1 KNOWLEDGE.....	18
2.2 KNOWLEDGE MANAGEMENT .....	22
2.3 KNOWLEDGE MANAGEMENT SYSTEMS.....	24
2.4 KNOWLEDGE MANAGEMENT PROCESSES .....	27
2.4.1 <i>Knowledge Creation</i> .....	27
2.4.1.1 Information Technology Support of Knowledge Creation .....	30
2.4.2 <i>Knowledge Storage/Retrieval</i> .....	31
2.4.2.1 Information Technology Support of Knowledge Storage and Retrieval.....	31
2.4.3 <i>Knowledge Transfer</i> .....	32
2.4.3.1 Information Technology Support of Knowledge Transfer.....	33
2.4.4 <i>Knowledge Application</i> .....	33
2.4.4.1 Information Technology Support of Knowledge Application .....	34
2.4.5 <i>Summary of the Knowledge Management Processes</i> .....	35
2.5 KNOWLEDGE MANAGEMENT: STRATEGIES AND FRAMEWORKS .....	36
2.5.1 <i>Knowledge Flows</i> .....	39
2.5.2 <i>Knowledge Sharing</i> .....	41
2.6 CONCLUSION .....	43
<b>CHAPTER 3 KNOWLEDGE WITHIN PROJECTS .....</b>	<b>44</b>
3.1 WHAT ARE PROJECTS?.....	44
3.2 PROJECT MANAGEMENT .....	45
3.3 PROJECT MANAGEMENT MATURITY .....	46
3.4 KNOWLEDGE USE IN PROJECTS .....	47
3.5 PROJECT TEAMS.....	51
3.6 CONCLUSION .....	52
<b>CHAPTER 4 KNOWLEDGE IN SOFTWARE DEVELOPMENT PROJECTS.....</b>	<b>54</b>
4.1 THE NEED FOR KNOWLEDGE IN SOFTWARE PROJECTS .....	54
4.2 MANAGING KNOWLEDGE IN SOFTWARE PROJECTS.....	55
4.3 SOFTWARE PROJECT ORGANISATION MATURITY.....	57
4.4 CONCLUSION .....	59
<b>PART II RESEARCH METHODOLOGY .....</b>	<b>60</b>
<b>CHAPTER 5 DEFINING THE RESEARCH PROBLEM .....</b>	<b>61</b>
5.1 HYPOTHESIS AND RESEARCH QUESTION.....	61
5.2 THE DYNAMIC FEEDBACK MODEL .....	62
5.3 ADAPTING THE DFM .....	65
5.3.1 <i>Decision-Making Area and PDCA Plan Step</i> .....	67
5.3.2 <i>Technical Development and PDCA Do Step</i> .....	68
5.3.3 <i>Quality Assurance and PDCA Check Step</i> .....	68
5.3.4 <i>Project Management and PDCA Act Step</i> .....	69
5.3.5 <i>Summary – Adapting the DFM</i> .....	70
5.4 CONCLUSION .....	70
<b>CHAPTER 6 RESEARCH FRAMEWORK.....</b>	<b>72</b>
6.1 AN OVERVIEW OF RESEARCH METHODS .....	72
6.2 RESEARCH DESIGNS.....	75
6.2.1 <i>Experimental Research</i> .....	75
6.2.2 <i>Survey Research</i> .....	75
6.2.3 <i>Case Study</i> .....	76
6.2.4 <i>Action Research</i> .....	77
6.3 METHODS OF DATA COLLECTION .....	78
6.3.1 <i>Observation</i> .....	78

6.3.2 Interviews.....	79
6.3.3 Questionnaire.....	79
6.3.4 Documentation and Archival Records.....	80
6.4 RESEARCH IN INFORMATION SYSTEMS AND COMPUTING.....	81
6.5 RESEARCH DESIGN.....	82
6.5.1 The Case Study Organisation.....	82
6.5.2 Structuring the Research.....	83
6.5.3 Fitting Knowledge Management within Project Management and Software Development Processes.....	86
6.5.4 Data Collection and Analysis within the Research Design.....	87
6.6 CONCLUSION.....	89
<b>CHAPTER 7 DATA COLLECTION AND FINDINGS.....</b>	<b>91</b>
7.1 DATA COLLECTION.....	91
7.1.1 Interviews.....	92
7.1.2 Survey Questionnaire.....	95
7.1.3 Observation.....	96
7.1.4 Constraint during Data Collection.....	99
7.1.5 Summary of Data Collection Activities.....	100
7.2 FINDINGS.....	100
7.2.1 Data Reduction and Display.....	102
7.2.2 Analysis and Conclusion Drawing.....	110
7.3 CONCLUSION.....	111
<b>PART III EVALUATION, ANALYSIS AND KM CAPABILITY FRAMEWORK.....</b>	<b>112</b>
<b>CHAPTER 8 EVALUATION AND SENSE-MAKING.....</b>	<b>113</b>
8.1 KNOWLEDGE VISION.....	113
8.2 XYZ'S BODY OF KNOWLEDGE.....	114
8.3 KNOWLEDGE STRATEGY.....	115
8.4 THE KNOWLEDGE INITIATIVE.....	117
8.4.1 Issues addressed while implementing the knowledge initiative.....	118
8.4.2 Process Assets Library.....	122
8.4.3 Tacit Knowledge Exchange and Flow.....	123
8.4.4 The Knowledge Map for People (KM-P).....	126
8.5 INDIVIDUAL ROLES.....	127
8.6 ISSUES FACED BY XYZ AS A DISTRIBUTED ORGANISATION.....	129
8.7 MEASURES.....	130
8.8 CONCLUSION.....	131
<b>CHAPTER 9 VALIDATING THE DYNAMIC FEEDBACK MODEL.....</b>	<b>132</b>
9.1 FUNCTIONAL AREAS AND THEIR KNOWLEDGE FLOWS.....	132
9.1.1 Technical Development Area.....	133
9.1.2 Quality Area.....	134
9.1.3 Project Management Area.....	135
9.1.4 Decision-Making Area.....	140
9.1.5 Summary of Functional Areas.....	141
9.2 RELATIONSHIP BETWEEN KNOWLEDGE FLOWS AND FUNCTIONAL AREAS.....	141
9.2.1 Interactions of the Relationships between Knowledge Flows and Functional Areas.....	144
9.3 FEEDBACK LOOPS.....	145
9.3.1 Knowledge Creation--Learning--Application Loop.....	145
9.3.2 Knowledge Creation--Reflection--Transfer--Application Loop.....	147
9.3.3 The Experience--Transfer--Application Loop.....	148
9.3.4 Summary of Feedback Loops.....	149
9.4 KNOWLEDGE FLOWS, FUNCTIONAL AREAS, DFM AND THE K-DFM.....	150
9.4.1 Evaluating K-DFM with Grady's PDCA Software Process Improvement Cycle.....	151
9.4.2 Assessing the K-DFM-- Revisiting Rubenstein-Montano et al's Criteria for Knowledge Management Frameworks.....	153
9.4.3 Summary of Functional Areas, Knowledge Flows and K-DFM.....	155
9.5 CONCLUSION.....	156
<b>CHAPTER 10 KM CAPABILITY FRAMEWORK.....</b>	<b>157</b>
10.1 KM CAPABILITIES.....	157
10.1.1 KM Infrastructure Capabilities.....	158
10.1.2 KM Process Capabilities.....	159

10.2 DEVELOPMENT OF KM CAPABILITIES AT XYZ.....	159
10.2.1 <i>The Initial State</i> .....	160
10.2.2 <i>The Deviation</i> .....	162
10.2.3 <i>The Correction</i> .....	164
10.3 KM CAPABILITY FRAMEWORK.....	166
10.3.1 <i>Initial State</i> .....	167
10.3.2 <i>High KM Infrastructure Capability</i> .....	168
10.3.3 <i>High KM Process Capability</i> .....	169
10.3.4 <i>Organisational State of KM</i> .....	169
10.3.5 <i>Framework Summary</i> .....	170
10.4 CONCLUSION.....	170
<b>PART IV CONCLUSIONS.....</b>	<b>171</b>
<b>CHAPTER 11 CONCLUSIONS.....</b>	<b>172</b>
11.1 KEY POINTS.....	172
11.1.1 <i>The Knowledge Dynamic Feedback Model (K-DFM)</i> .....	173
11.1.2 <i>Directory of Experts</i> .....	174
11.1.3 <i>Knowledge Champions in a Globally Distributed Software Organisation</i> .....	174
11.1.4 <i>Developing Knowledge Management Capabilities</i> .....	175
11.1.5 <i>Summary of Key Points</i> .....	176
11.2 CONTRIBUTION.....	177
11.3 LIMITATIONS.....	180
11.4 IMPLICATIONS.....	182
11.5 FUTURE WORK.....	185
11.6 CHAPTER SUMMARY.....	186
<b>II. REFERENCES.....</b>	<b>187</b>
<b>APPENDIX 1 KNOWLEDGE MANAGEMENT FRAMEWORKS.....</b>	<b>204</b>
<b>APPENDIX 2 PROJECT TEAMS, COMMUNITY OF PRACTICE, AND BA.....</b>	<b>207</b>
<b>APPENDIX 3 REFEREED AND PUBLISHED PROCEEDINGS.....</b>	<b>208</b>
3.1. DALCHER, D AND SANDHAWALIA, B, "THE DYNAMIC PERSPECTIVE: INCORPORATING KNOWLEDGE AND LEARNING INTO THE SOFTWARE DEVELOPMENT PROCESS", PROCEEDINGS SOFTWARE QUALITY MANAGEMENT (SQM) XII -- NEW APPROACHES TO SOFTWARE QUALITY, BRITISH COMPUTER SOCIETY, CANTERBURY, UK, MARCH-APRIL 2004.....	208
3.2. DALCHER, D AND SANDHAWALIA, B, "LEARNING AND KNOWLEDGE CAPTURE IN PROJECTS", PROCEEDINGS 5TH EUROPEAN CONFERENCE ON KNOWLEDGE MANAGEMENT (ECKM), CNAM, PARIS, FRANCE, SEPTEMBER-OCTOBER, 2004.....	208
3.3. SANDHAWALIA, B AND DALCHER, D, "NEGOTIATION AND DECISION-MAKING WITHIN SOFTWARE PROJECTS", PROCEEDINGS GROUP DECISION AND NEGOTIATION (GDN) ANNUAL CONFERENCE, VIENNA, AUSTRIA, JULY 2005.....	208
3.4. SANDHAWALIA, B AND DALCHER, D, "KNOWLEDGE MANAGEMENT SUPPORT FOR SOFTWARE PROJECT ORGANISATIONS", PROCEEDINGS 4TH ANNUAL CONFERENCE ON INFORMATION SCIENCE, TECHNOLOGY AND MANAGEMENT (CISTM), CHANDIGARH, INDIA, JULY 2006.....	209
3.5. SANDHAWALIA, B AND DALCHER, D, "KNOWLEDGE AND DECISION-MAKING WITHIN SOFTWARE PROJECTS", PROCEEDINGS 15TH INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS DEVELOPMENT (ISD), BUDAPEST, HUNGARY, AUGUST-SEPTEMBER, 2006.....	209
3.6. SANDHAWALIA, B AND DALCHER, D, "KNOWLEDGE SUPPORT FOR SOFTWARE PROJECTS", PROCEEDINGS 18TH ANNUAL INFORMATION RESOURCES MANAGEMENT ASSOCIATION (IRMA) CONFERENCE, VANCOUVER, CANADA, MAY 2007.....	210
3.7. SANDHAWALIA, B AND DALCHER, D, "KNOWLEDGE MANAGEMENT CAPABILITY FRAMEWORK", ACCEPTED PAPER AT KNOWLEDGE MANAGEMENT IN ACTION (KMIA) CONFERENCE TO BE HELD IN MILAN, ITALY, SEPTEMBER 7-10, 2008.....	210
<b>APPENDIX 4 SAMPLE INTERVIEWS.....</b>	<b>211</b>
<b>APPENDIX 5 SAMPLE QUESTIONNAIRE.....</b>	<b>238</b>

## List of Tables and Figures

Figure 1 Knowledge Creation Spiral.....	29
Figure 2 Project Management Knowledge.....	48
Figure 3 Lost Project Knowledge.....	49
Figure 4 Dynamic Feedback Model.....	64
Figure 5 Hypothesis: Adapted DFM reflecting Knowledge.....	67
Figure 6 Research Design.....	90
Figure 7 Inflow and Outflows for Technical Area.....	142
Figure 8 Inflow and Outflows for Quality Area.....	142
Figure 9 Inflows and Outflow for Project Management Area.....	143
Figure 10 Inflows and Outflow for Decision-Making Area.....	143
Figure 11 Knowledge Flow within and between Functional Areas.....	145
Figure 12 Knowledge Creation—Learning—Application.....	146
Figure 13 Knowledge Creation—Reflection—Transfer—Application Loop.....	147
Figure 14 Experience—Transfer—Loop.....	149
Figure 15 KM Capabilities.....	160
Figure 16 KM Infrastructure Development.....	162
Figure 17 KM Process Capability Development.....	164
Figure 18 Towards an Organisational State of KM Capability Development.....	165
Figure 19 KM Capability Framework.....	167
Table 1 Difference between KM and Information Management.....	26
Table 2 Interviews.....	93
Table 3 Questionnaire.....	96
Table 4 Summary of Observation Hours.....	98
Table 5 Observation Hours per Week.....	99
Table 6 Data Collected from Ongoing Projects.....	101
Table 7 Data Reduction and Display.....	104
Table 8 Categories and Attributes.....	105
Table 9 Relationship between Knowledge Flows and Functional Areas.....	144
Table 10 Evaluating K-DFM with Grady's PDCA Cycle.....	152
Table 11 Assessing K-DFM against Knowledge Management Framework Criteria.....	155
Table 12 Contribution of Thesis.....	180

## I. Abstract

Projects are unique by definition. Due to this novelty software development projects, in common with all other projects, require knowledge for effective implementation. Most knowledge management frameworks reported in the literature address the organisational need to manage knowledge. The existing frameworks typically discuss the dichotomy between tacit and explicit knowledge, and lay an emphasis on managing the latter. However, software development projects rely upon the experience, creativity and intuition of individual team members to address unstructured situations typified by inherent uncertainty, ambiguity and change. Therefore software projects require the facilitation and interaction of tacit knowledge along with managing and leveraging explicit knowledge. This research examines how tacit and explicit knowledge generated while implementing a software development project can be leveraged and effectively reused in future software projects.

In order to address the need to provide knowledge management support to software projects an extended case study was conducted at one of the world's largest software project-based organisations. The aim of the research was to identify and analyse the flow of knowledge, and the capabilities required to support this flow. The research design utilised a combination of open-ended interviews, survey questionnaires, observations of team functioning, work methods and development practices, and a detailed examination of the knowledge management infrastructure and process capabilities. The extensive and exceptional access negotiated for this project enabled the research to focus on a single organisation and resulted in 100 hours of interviews and 340 hours of observations from 98 ongoing projects. Established case study protocols were used for data collection. The data analysis focused on determining categories from the different streams of activities and assigning attributes using Nudist software for data reduction and displaying group-nodes, and conclusion drawing. This enabled the research to establish the 'processual' nature of knowledge, and identify the capabilities required to mobilise and utilise knowledge assets.

The research critically analysed the three parallel themes of knowledge management, project management and software engineering, and the outcome of the conceptual synthesis and validation is a dynamic model which represents the knowledge processes that facilitate the flow of tacit and explicit knowledge between software projects. The model depicts the relationships and interactions between the functional areas of the development



effort, and presents a continuous and long-term view of supporting the implementation of software projects and developing knowledge practices. For software project-based organisations this research has implications for their ability to manage context, provide feedback and facilitate interaction, and thus build upon their existing knowledge resources and capabilities. The research provides such organisations with a perspective to achieve excellence not only through optimisation of software process improvement, but also through learning, and, the creation and sharing of tacit and explicit knowledge as facilitated by the proposed model.

## Acronyms

1. AAR – After Action Review
2. BOK – Body of Knowledge
3. CASE – Computer Aided Software Engineering
4. CBR – Case Based Reasoning
5. CMM – Capability Maturity Model
6. CMMI – Capability Maturity Model Integration
7. CRM – Customer Relationship Management System
8. CSI – Customer Satisfaction Index
9. CSCW – Computer Supported Cooperative Work
10. DEF – New Knowledge Management System at Case Study Organisation
11. DFM – Dynamic Feedback Model
12. EKMS – Electronic Knowledge Management System
13. ERD – Entity Relationship Diagram
14. HR – Human Resources
15. IPMS – Integrated Project Management System
16. K-DFM – Knowledge Dynamic Feedback Model
17. KM – Knowledge Management
18. KM-P – Knowledge Map for People
19. MATC – Manpower Allocation Task Committee
20. PAL – Process Assets Library
21. PDCA – Plan/Do/Check/Act
22. PL – Project Lead
23. PM – Project Manager
24. PIP – Process Improvement Proposals
25. PMI – Project Management Institute
26. PMBOK – Project Management Body of Knowledge
27. QAG – Quality Assurance Group
28. QMS – Quality Management System
29. SEPG – Software Engineering Process Group
30. XYZ – Case Study Organisation

## Chapter 1 Introduction

Managing projects has become a well accepted way to manage organisations and implement strategy as organisations adopt a project approach to carry out a range of vital operational and innovative activities. Projects are temporary by nature and their implementation requires creative actions, practitioner's experience, and the ability to apply existing knowledge to development problems. The amount of knowledge required to manage a project depends upon the novelty and uniqueness of the required outcome. Love et al (1999) argue that even though a project is unique, the processes involved in delivering the final outcome are similar in project organisations and, therefore, most projects do not need to start from scratch as they can utilise existing processes and learn from the experiences acquired from previous projects. Projects are required to be completed within a specific schedule and budget, which makes the reuse and harnessing of knowledge desirable. Without the reuse of existing knowledge or the ability to create new knowledge from existing solutions and experiences, project organisations have to create new solutions to every problem they encounter, potentially leading to delays and inefficiencies. With the reuse of knowledge, project organisations can more efficiently plan to deliver projects within budget and on time. The effectiveness of reusing knowledge would be dependent upon the mechanisms and processes for learning and knowledge reuse within a project's life-cycle.

Also, while implementing a project, effort is often focused on immediate deliverables with no emphasis on how the experience and insights gained would help and benefit future projects. Compared with non-project organisations that can be characterised as supported organisational structures and knowledge-absorbing routines, where knowledge is routinised and socialised into the organisation, project-based organisations are generally considered not to have support mechanisms that enable knowledge transfer to occur. Project-based organisations work on time-scales that are typically organised around teams assembled specifically for the project and often disbanded upon its completion. Requirements evolve and team members often change during the course of projects, while feedback from one phase of the project to another rarely provides team members with an opportunity to learn from their good decisions or mistakes. Team members often come together for the first time at the outset of the project and therefore it is difficult to create the right knowledge culture and locate the knowledge assets. Moreover, project organisations

are normally dispersed and fragmented horizontal structures making it difficult to effectively transfer knowledge and project learning across project boundaries.

Software development projects are life cycle driven and typically follow the sequence of going from concept through definition and development, to testing and delivery, Dalcher (2002a and 2002b). However, unlike other projects, the requirements of software projects are subject to frequent change. As a product, software can be changed, and it is therefore assumed that change is possible even during the later stages of the development process. Change and uncertainty can combine to make software projects more unpredictable than other projects. Myers (1985) states that more than half the cost of complex software development is attributable to decisions made in the 'upstream' portion, or initial stages of the development process, namely, requirements specification and design. Traditional software development models like the waterfall (Royce 1970) and spiral (Boehm 1988) follow a sequential approach where the 'upstream' and 'downstream' aspects of development are clearly distinguishable. However, modern approaches like agile methods are more emergent and evolutionary, and rely on frequent feedback and interaction between and within self-organised teams while attempting to address change and uncertainty in the requirements. Knowledge is the raw material required for decision making for all projects, particularly within these self-organised software teams. For complex projects, knowledge from multiple technical and functional domains is required (Curtis et al 1988). This knowledge, combined with the knowledge generated while executing the project and the lessons learnt in the wake of the project, make a valuable knowledge asset that needs to be effectively leveraged for future projects. Lindvall and Rus (2003) state that 'since individuals are the ones developing software, the ultimate goal is for them to have access to the right knowledge at the right time. Thus, new knowledge might be acquired, and existing individual knowledge must be leveraged to the organisational level and then distributed back to the individuals who need it.'

The implementation and outcome of a project depends to a large extent on the knowledge of individuals, their access to local and global knowledge resources, and recognition and integration of existing knowledge. Problem solving within unique project instances generates further knowledge, and the knowledge assets thus created, combined with the experience gained by implementing the project, can benefit future projects. Enhanced business value can be gained with future projects delivered on schedule and within budget by making the implementation more efficient and effective. Certain software process improvement approaches, for example the Capability Maturity Model (SW-CMM), suggest

that the development process be optimised to deliver the most of the software organisation's capability. Such approaches often suggest that knowledge can be managed or leveraged, but do not address the specific issues relevant to the knowledge requirements of the project. The knowledge requirements make it imperative to identify what knowledge needs to be managed, how, when, where, by whom, and for whom. Consequently, the key requirements for managing knowledge within software projects are: collecting and organising the knowledge; making it available through knowledge infrastructure; and using the knowledge to improve the execution of projects. To address these key requirements within the software development project perspective, this research investigated the following:

- How is knowledge generated and where?
- How is knowledge shared?
- How is knowledge managed, codified, transferred, reused and applied?
- How is knowledge assimilated within the software project organisation?
- How does knowledge flow within the processes of a software project organisation, and what infrastructure is required to support this flow?
- How does knowledge occur within the different levels of a software project organisation – individuals, teams, project and organisation?

Time is a key factor while differentiating a knowledge management (KM) initiative in a project from an organisational initiative. An organisation aims at establishing a KM initiative as a permanent process in its operations. KM in a project is a process with specific start and end dates, and clear purpose to support planning, implementation and monitoring of the project. As such, the opportunity and time available to capture the knowledge generated within a project is limited. Therefore a project organisation needs to adopt a continuous and long term perspective towards benefiting from knowledge generated while executing a project and leveraging that knowledge to effectively implement future projects.

This research presented a previously published theoretical model that provides feedback between the functional areas of project management, decision-making, technical development and quality assurance. The research further adapted this framework to incorporate the knowledge processes that underpin and support the project management and software development processes. The research analysed how the feedback loops of the published model facilitate interaction between the functional areas and enable the flow of knowledge for effective implementation of the software development effort. The adapted

version of the model is mapped and analysed against Grady's (1997) PDCA cycle for software process improvement to evaluate its functionality and ability to make software processes more effective.

To validate the model the research conducted a case study at a large CMMI Level 5 software project organisation that has over 150 offices worldwide in more than 40 countries and employs in excess of 108,000 individuals, and is still growing. The organisation is a leading global provider of software applications and solutions with a reputation for excellence. Exceptional access negotiated for this research project provided an opportunity to study and analyse the well established and highly mature work methods and practices of the organisation's software development effort. The research propositions required a study of how a software project organisation's knowledge management initiative supports project management and software development processes. The researcher was able to observe the project management, knowledge and software process within a highly mature organisation, and the functioning of project teams in their work environment. In addition to this, the researcher was able to interview a cross-section of individuals at the organisation that included an Executive Vice President, members of the SEPG, Principal Consultants, Project Managers and software developers. The researcher was also able to administer a questionnaire to receive feedback from key members based at client sites.

The case study provided evidence of the flow of knowledge and how individuals work while implementing software projects. Based upon the empirical evidence gained while conducting the case study the research was able to establish feedback and interaction as the key elements that facilitate the flow of knowledge, both tacit and explicit, within the functional areas and consequently across the organisation.

The validated model was also assessed against Rubenstein-Montano et al's (2000) criteria for knowledge management frameworks. The assessment established that the model: is dynamic and is consistent with systems thinking; links knowledge management to a software project organisation's goal and strategy of continuous process improvement; provides knowledge management support to project management and software development processes; acknowledges organisational culture and provides the framework to facilitate interaction and knowledge sharing; and, enables the flow of knowledge and learning through feedback loops.

Facilitating the flow of knowledge becomes more critical in a distributed environment and requires knowledge management capabilities to maintain and support such a flow. The research examined the capability support required to facilitate the flow of knowledge across a distributed organisation, and also presents a framework that enables organisations to analyse and monitor the progress of their KM capabilities development, and thus ensure successful implementation of a KM initiative.

The research established how software organisations design and organise their work practices, and the capabilities required to support the activities of the development effort. Thus this research provides a new perspective and better understanding of the interactions and relationship between software development, project management and knowledge management processes within the work practices of a globally distributed software organisation. By facilitating the flow of knowledge within the functional areas, this research presents a new and novel way of viewing how to ensure that the right knowledge is available to the right person at the right time during the software development effort. The following section discusses the originality of this research.

## **1.1 Originality of Work**

Cryer (2000) examines originality in research and provides examples and outcomes. Such examples and outcomes of original research include a new or improved model or perspective, as well as a critical analysis or an in-depth study and exploration of a topic, area or field. This research examines and provides a conceptual understanding of how knowledge management processes support project management and software development processes. The research adopts a project management perspective of software development, and provides a knowledge management framework to effectively implement a KM initiative within a software project organisation. To achieve this, the research adapts and extends a published model framework that depicts the continuous nature of software development in contrast to the typically static representation of the process. In doing so it offers an extension to a published model, an improved perspective and the validation of the utility of the model in a practical setting. The research evaluates and assesses how knowledge management processes can be facilitated by the model to support software projects. While previous frameworks present in the literature support knowledge management within permanent organisational structures and are considered either prescriptive or descriptive, this model provides a framework that is extended to facilitate

the dynamic nature and flow of knowledge and its management within project environments. The research also presents a separate framework that enables organisations to analyse their KM capabilities development and identify any imbalance, and therefore make corrections if required to ensure successful implementation of a KM initiative.

The research analyses and synthesises the literature, theories, principles and processes from the disciplines of knowledge management, information systems, software engineering and project management. The research presents a new perspective about the relationship and interactions between these disciplines and how knowledge management processes support software development and project management processes. The research adapts, extends and validates a model published and accepted in literature that had previously not been validated. The adapted and extended version of the model is applied to facilitate feedback and interaction and facilitate knowledge within functional areas, and validated based upon the empirical evidence of the case study conducted.

The research also presents a framework that maps the progress and development of the two dimensions of KM capabilities, namely infrastructure and processes, towards an organisational state where they are embedded within the activities, routines, procedures and work practices of a project organisation. The framework benefits organisations as they are able to monitor the progress of their capability development to ensure successful implementation of their KM initiatives. This framework is also based upon the empirical evidence and insights gained while conducting an in-depth case study in a large software organisation. The research therefore provides a better understanding of the work practices of a globally distributed software organisation with regard to the use of knowledge to effectively implement software projects. The following section introduces the structure of the remaining chapters of the thesis.

## **1.2 Organisation of thesis**

The thesis is organised in four parts. Part 1 of the thesis contains three chapters. Chapter 2 provides the background for the research and creates the context within which the work is situated. The chapter reviews the current literature on knowledge, knowledge management and its systems, processes, strategies and frameworks, and how these issues relate to the organisational context. Chapter 3 discusses how projects differ from organisations, and how the issues related to knowledge management become even more critical under project



conditions. The chapter also discusses the different types of knowledge required by project teams to effectively implement projects. Chapter 4 presents the need for knowledge in software projects, and the strategies that exist within the literature to manage and leverage this knowledge. The chapter also discusses how optimised and standardised software processes, as recommended by the Capability Maturity Model (CMM), influence the knowledge management issues of a software organisation.

Part 2 of the thesis also contains three chapters that relate to the methodology of this research. Chapter 5 defines the research problem, and presents the hypothesis and research question. The chapter discusses the research issues to be examined, and formulates the hypothesis for the research as an adaptation of a published model. The model is adapted to incorporate the flow of knowledge, feedback and interaction. Chapter 6 presents an overview of research methods and the research design for this study, and discusses the methods adopted to achieve the research objectives. Chapter 7 discusses how the data was collected and initially analysed while conducting the case study.

Part 3 of the thesis also contains three chapters that present the findings of the case study. Chapter 8 discusses the evaluation and insights gained by analysing the data collected during the case study regarding the organisation's knowledge vision, strategy, initiative, flow and issues addressed while developing the knowledge management initiative. Chapter 9 validates the research question and hypothesis, by analysing the flow of knowledge between the software project organisation's functional areas. Chapter 10 presents an analysis of the knowledge management infrastructure and process capabilities of the software project organisation. Based upon this analysis, the chapter develops and presents a framework to analyse the development of knowledge management infrastructure and process capabilities.

Part 4 of the thesis contains one chapter that presents the conclusions and contribution of the research. The chapter also discusses the limitations and implications of the study, and the possibilities and scope of future work based upon the findings and conclusions of this research.

## Part I Literature Review

## Chapter 2 Understanding Knowledge and its Management

This chapter provides the background for the research and creates the context within which the work is situated. It reviews the current literature on knowledge, knowledge management and its systems, processes, strategies and frameworks. The chapter discusses how information technology facilitates and supports knowledge management initiatives. It also highlights the importance in identifying the flow of knowledge and how it is shared. The discussion relates to knowledge management initiatives in the organisational context, which provides the basis and motivation to view these issues within projects, and software development projects in particular, as presented in the subsequent chapters. The chapter begins by addressing ‘what is knowledge?’

### 2.1 Knowledge

The Oxford English Dictionary (2002) states that knowledge is:

- i) information or skills acquired through experience or education
- ii) the sum of what is known

Philosophically, the dictionary continues, knowledge is:

- i) justified true belief, as opposed to opinion
- ii) awareness or familiarity gained by experience

The dictionary’s statement depicts the essence of knowledge which is often perceived as:

- The state or fact of knowing
- Familiarity, awareness or understanding gained through experience or study
- The sum or range of what has been perceived, discovered or learned
- Specific information about something

Knowledge is a mixture of various elements and is considered to be about beliefs, meaning, intention, commitment, action, perspective and context. Knowledge is also believed to be fluid and formally structured, and it exists within people, processes, structures and routines. Davenport and Prusak (1998) attempt to capture this essence of knowledge in their following definition:

“Knowledge is a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and information. It originates and is applied in the minds of knowers. In organisations, it often becomes embedded not only in documents or repositories but also in organisational routines, processes, practices, and norms.”

The importance of knowledge is increasing as organisations recognise that they possess knowledge and view this knowledge as a valuable asset. Knowledge is embedded in and carried through multiple entities including organisation culture and identity, routines, policies, systems, documents and individuals, Grant (1996) and Spender (1996). In attempting to capitalise on the competitive advantage provided by this knowledge, organisations try to leverage it by using information technology software such as databases and electronic conferencing to facilitate the acquisition, sharing, storage, retrieval and application of the knowledge. However, information technology processes data and information which leads to the hierarchical view of data, information and knowledge where data is raw numbers and facts, information is processed data, and knowledge is authenticated information, Dretske (1981), Machulp (1983), and Vance (1997).

Alavi and Leidner (2001) posit that information is converted to knowledge once it is processed in the mind of individuals and knowledge becomes information once it is articulated and presented in the form of text, graphics, words, or other symbolic forms. A significant implication of this view of knowledge is that for individuals to arrive at the same understanding of data or information, they must share a certain knowledge base. The shared knowledge base is important in the exchange of knowledge, especially in the interaction of tacit and explicit knowledge. Tacit knowledge comprises an individual's mental models; is personal and in the mind of an individual; is context specific and difficult to articulate, formalise and verbalise; and is therefore hard to communicate. The factors that influence an individual's mental model include the individual's education, expertise, past experiences, perceptions, biases, prejudices and environment. Explicit knowledge can be easily articulated and codified and therefore transmitted and communicated. Polanyi (1967) contends that human beings acquire knowledge by actively creating and organising their own experiences and sums it up by stating that “we can know more than we can tell.”

Polanyi's (1967) influence is evident in contemporary discussion about organisational knowledge or the collective knowledge available within an organisation. Organisational

knowledge includes the knowledge which resides within the individuals, systems, processes, documents and structures of the organisation. Tacit knowledge is important for competitive advantage as it is the unexpressed knowledge and experiences of an organisation that provide the unique competencies that cannot be replicated by competitors, Barney (1991). While tacit and explicit knowledge are part of an individual's and organisation's knowledge, Cook and Brown (1999) propose that knowledge needs to be a 'part of action,' that is, it needs to be practiced. To account for all the knowledge of an entity Cook and Brown (1999) propose that the notion of knowing is important, for while tacit and explicit knowledge are 'possessed' by an entity, knowing is not about possession but about practice and interaction. Wiig (1996) includes the tacit, explicit and knowing aspects of knowledge by defining it as "the insights, understandings, and practical know-how that we all possess – it is the fundamental resource that allows us to function intelligently."

According to the knowledge-based theory of the firm, the source of competitive advantage resides in the ability of an organisation to turn knowledge into action and less on knowledge itself. Grant (1996) argues that the integration of knowledge, either explicitly or implicitly, of many different people to facilitate knowledge application is the motivation for organisations comprised of multiple individuals. Recognising that integration of knowledge of organisational individuals is exceptionally difficult, Grant (1996) advocates that a key challenge for organisations is to achieve effective knowledge application that facilitates the integration of an individual's specialist knowledge. Grant (1996) suggests that such integration of specialist knowledge can be facilitated by establish a mode of interaction amongst specialists within the organisation.

Hansen et al (1999) contend that various knowledge initiatives ignore the social architecture of knowledge exchange within organisations. Von Krogh et al (2000) are of the view that knowledge creation needs to be enabled and facilitated, and not managed. They underline the importance of managing organisational relationships, especially in the case of micro-communities of knowledge. Micro-communities of knowledge are similar to communities of practice, and are small groups within organisations whose members share what they know as well as common values and goals. Wenger and Snyder (2000) define communities of practice as "groups of people informally bound together by shared experience and passion for joint enterprise." Due to common values these communities evolve rather than being project or deadline driven, and develop when there are ample opportunities for informal contact. Over time, the interactions among individuals enable

them to develop their own routines and share assumptions, experiences and knowledge. Leonard and Sensiper (1998) suggest that a certain level of personal intimacy is necessary that allows the communication and transfer of tacit knowledge. The communities also evolve practices, and often create their own languages, which may contain jargon and colloquialisms whose meanings outsiders may not be familiar with. Such communities may or may not be a part of a team structure as organisational teams are often hierarchical in nature, and project teams are unified by a specific goal and deliverable, while communities of practice are informal groups that share knowledge about a common topic.

The different perspectives recognise knowledge to be a source of competitive advantage and innovation. As stated by Nonaka and Takeuchi (1995), “when organisations innovate, they do not simply process information, from the outside in, in order to solve existing problems and adapt to a changing environment. They actually create new knowledge and information, from the inside out, in order to redefine both problems and solutions and, in the process, to re-create their environment.” To explain innovation, Nonaka and Takeuchi (1995) propose their theory of knowledge creation, in which they view knowledge as ‘justified true belief,’ where they consider knowledge to be about beliefs, commitment, action and meaning -- in other words intention, practice and context.

Alavi and Leidner (1999) embody the above essence by defining knowledge as ‘a justified personal belief that increases an individual’s capacity to take effective action.’ While ‘personal’ implies the contextual nature of knowledge, action requires competencies and know-how, and implies the dynamic nature of knowledge. Nonaka and Takeuchi (1995) focus on the dynamic characteristic of knowledge, as in an organisation knowledge changes around products, services, processes, technology, structures, roles and relationships. They view knowledge creation as a dynamic process, as knowledge created becomes the basis for further knowledge creation. In their SECI model of knowledge creation, Nonaka and Takeuchi (1995) propose that “tacit knowledge of individuals is the basis of organisational knowledge creation.” They imply organisational knowledge creation to be a spiral process, starting at the individual level and moving up through expanding communities of interaction, that cross sectional, departmental, divisional and organisational boundaries. This organisational knowledge is viewed as a valuable strategic asset, and Davenport and Prusak (1998) recommend that to remain competitive, organisations must efficiently and effectively create, capture, locate and share their organisations knowledge and expertise, and have the ability to bring that knowledge to bear on problems and opportunities.

## 2.2 Knowledge Management

Organisations routinely engage in the generation, capture and use of knowledge in order to develop and deliver their products and services competitively. Information technologies have been closely associated with the development of such initiatives. Recently, the trend has been toward the application of advanced information technologies to systemise, facilitate and expedite firm-wide knowledge management. Alavi and Leidner (1999) refer to knowledge management as a systematic and organisationally specified process for acquiring, organising and communicating both tacit and explicit knowledge of employees so that other employees may make use of it to be more effective and productive in their work.

Knowledge management has been associated with the use of information technology systems that are intended to capture and distribute knowledge within organisations. This association with information technology has led to knowledge management being defined as “the explicit control and management of knowledge within an organisation aimed at achieving the company’s objectives,” Van der Spek and Spijkervet (1997); “the formal management of knowledge for facilitating creation, access and reuse of knowledge, typically using advanced technology,” O’Leary (1998); “the process of creating, capturing and using knowledge to enhance organisational performance,” Bassi (1999); and “the ability of organisations to manage, store, value and distribute knowledge,” Liebowitz and Wilcox (1997).

The field of knowledge management includes management and operational practices and philosophies, strategies and human behavioural issues, in addition to information technologies. Malhotra (1997) depicts the strategic view of knowledge management by considering the synergy between technological and behavioural issues. He states “knowledge management caters to the critical issues of organisational adaptation, survival and competence in the face of increasingly discontinuous environmental change. Essentially, it embodies organisational processes that seek synergistic combination of data and information processing capacity of information technologies, and the creative and innovative capacity of human beings.”

Knowledge management is increasingly being viewed as central to product and process innovation and improvement, and decision-making. This view is highlighted by Hackbarth (1998) who states that knowledge management is “purported to increase innovativeness

and responsiveness.” The American Productivity and Quality Centre (1996) further support this view by defining knowledge management as “a conscious strategy of getting the right knowledge to the right people at the right time and helping people share and put information into action in ways that strive to improve organisational performance.” Also, knowledge management is being viewed as the process by which organisations generate value through their knowledge assets. Von Krogh (1998) for example, endorses this view by referring to knowledge management as identifying and leveraging the collective knowledge in an organisation to help the organisation compete.

According to Davenport and Prusak (1998), most knowledge management projects have one of three aims:

- to make knowledge visible and show the role of knowledge in an organisation, mainly through maps, yellow pages, and hypertext tools
- to develop a knowledge-intensive culture by encouraging and aggregating behaviours such as knowledge sharing and proactively seeking and offering knowledge
- to build a knowledge infrastructure – not only a technical system, but a web of connections among people given space, time, tools and encouragement to interact and collaborate

To achieve these aims, organisational knowledge management initiatives adopt different perspectives of knowledge. An organisational view of knowledge as a process focuses on the flow of knowledge and the processes of creation, sharing and distribution of knowledge. An organisational view of knowledge as a capability results in developing competencies, locating the knowledge sources and creating intellectual capital. These different perspectives of knowledge result in different strategies of knowledge management adopted by organisations.

There are two basic strategies of knowledge management in which information technology provides support. The two strategies are: codification and personalisation, Hansen et al (1999). In the codification strategy, explicit and more structured knowledge is codified and stored in knowledge bases. Information technology helps individuals to share knowledge through common storage to enable knowledge reuse. The codification strategy relates to the content perspective that views knowledge as being able to be codified and stored in repositories, which allows for knowledge to be shared, built upon and retained regardless of employee turnover (Wasko and Faraj, 2000). This perspective focuses on collecting,



distributing, reusing and measuring existing codified knowledge (Cohen 1998, Knock and McQueen, 1998). In the personalisation strategy, tacit and more unstructured knowledge is shared through personal communication. Information technology helps individuals to locate each other and communicate effectively, thereby enabling knowledge transfer. Information technology achieves this knowledge transfer in a systemised manner across organisational and geographical boundaries, and in synchronous and asynchronous settings. The personalisation strategy relates to the relational perspective that views knowledge as being relative, provisional, context-bound, Orr (1990), Blacker et al (1993), and Barley (1996), and focuses on the process of knowing and the capability to act.

### **2.3 Knowledge Management Systems**

Davenport and Prusak (1998) note there are several reasons for knowledge workers not to apply their knowledge. Chief amongst these are social factors such as distrusting the source of knowledge or lack of time or opportunity to apply knowledge, Alavi and Leidner (2001). Observing that organisations tend to have a gap between what they know and what they do, Pfeffer and Sutton (2000), Alavi and Leidner (2001) suggest that information technology can have a positive influence on knowledge application. For example, information systems can enhance knowledge application by facilitating the capture, updating, and accessibility of organisational information and knowledge, Mao and Benbasat (1998). Also, information system systems can increase the size of knowledge workers' internal social networks by allowing for organisational knowledge to be applied across time and space, Kock and McQueen (1998).

Information technology is considered to play an important role within knowledge management systems. Alavi and Leidner (2001) refer to knowledge management systems as a class of information systems applied to managing organisational knowledge. In other words, they are information technology-based systems developed to support and enhance the organisational processes of knowledge creation, storage/retrieval, transfer and application. Knowledge management systems are designed to support and augment organisational knowledge management, and to complement and enhance the knowledge management activities of individuals. The broad range of activities supported by knowledge management systems include:

- creation of knowledge bases -- best practices, expert directories, market intelligence

- effective information management – gathering, filtering, classifying and storing
- incorporating knowledge into business processes
- development of knowledge centres – focal points of knowledge skills, mapping of internal expertise, and facilitating knowledge flows
- Reuse of knowledge – like customer support via case-based reasoning
- Enabling collaboration through technologies like groupware and intranets
- Creating knowledge webs and networks, and communities of practice beyond or across organisational boundaries
- Augmentation of decision support processes through expert systems or group decision support systems

While information technology is an enabler and facilitator of knowledge management, implementing an information technology infrastructure does not always result in a knowledge management initiative. The distinction between knowledge and information in the hierarchical view of knowledge leads to the further distinction between knowledge management and information management. Stear and Bair (1997) present the differences between the two in the following table (Table 1):

**Table 1 Difference between KM and Information Management**

<b>KM Characteristics</b>	<b>IM Characteristics</b>
Focus on capturing tacit and explicit information	Focus on recording and processing explicit information
Takes information from one source and promotes reuse in other situations	Takes information from multiple sources and organises it into a data-base system
Designed for distributed access, storage and control (end-user empowerment)	Designed for centralised information storage (IS empowerment)
Emphasis is on collaboration and sharing	Emphasis is on inquiries to highly structured repositories
Enables end-user defined information, relationships and needs	Concerned with information collection, classification and distribution
Employs technologies (e.g., visualisation) for knowledge discovery	Dependent on well-defined inquiries for retrieval
Adds value for growth, innovation and leverage	Required to maintain mission-critical enterprise data
Productivity for innovation	Productivity for efficiency

The different characteristics of knowledge management and information management determine the manner and perspective with which the technologies and tools are deployed. Information technology facilitates the creation of repositories of information through its classification, codification, storage, transfer and communication. Information technology therefore focuses on the core processes that enable organisations to function effectively. Knowledge management requires information technology to be complemented by a culture of sharing. A knowledge sharing culture ensures that this information is processed in the minds of the individuals as knowledge. Knowledge management focuses beyond the day-to-day functions and seeks to develop an organisational culture that enhances its individuals' capabilities and competencies. In attempting to do so, knowledge management seeks to effectively leverage and harness information technology to achieve the goal of maximising organisational knowledge through the processes of knowledge management. The following section will review the role of information technology in the knowledge management processes, in order to establish how information technology facilitates the creation, storage, transfer and application of knowledge.

## **2.4 Knowledge Management Processes**

Pentland (1995) defines knowledge management processes as “an ongoing set of practices embedded in the social and physical structure of the organisation with knowledge as their final product.” Alavi and Leidner (2001) develop a framework for knowledge management processes based upon the view of organisations as “knowledge systems.” According to this framework, organisations consist of four “knowledge processes”: creation, storage/retrieval, transfer and application. Alavi and Leidner (2001) also contend that information technology support is essential for effective knowledge processes. The following sections review each of the four knowledge processes in turn, and the role of information technology support for each process.

### **2.4.1 Knowledge Creation**

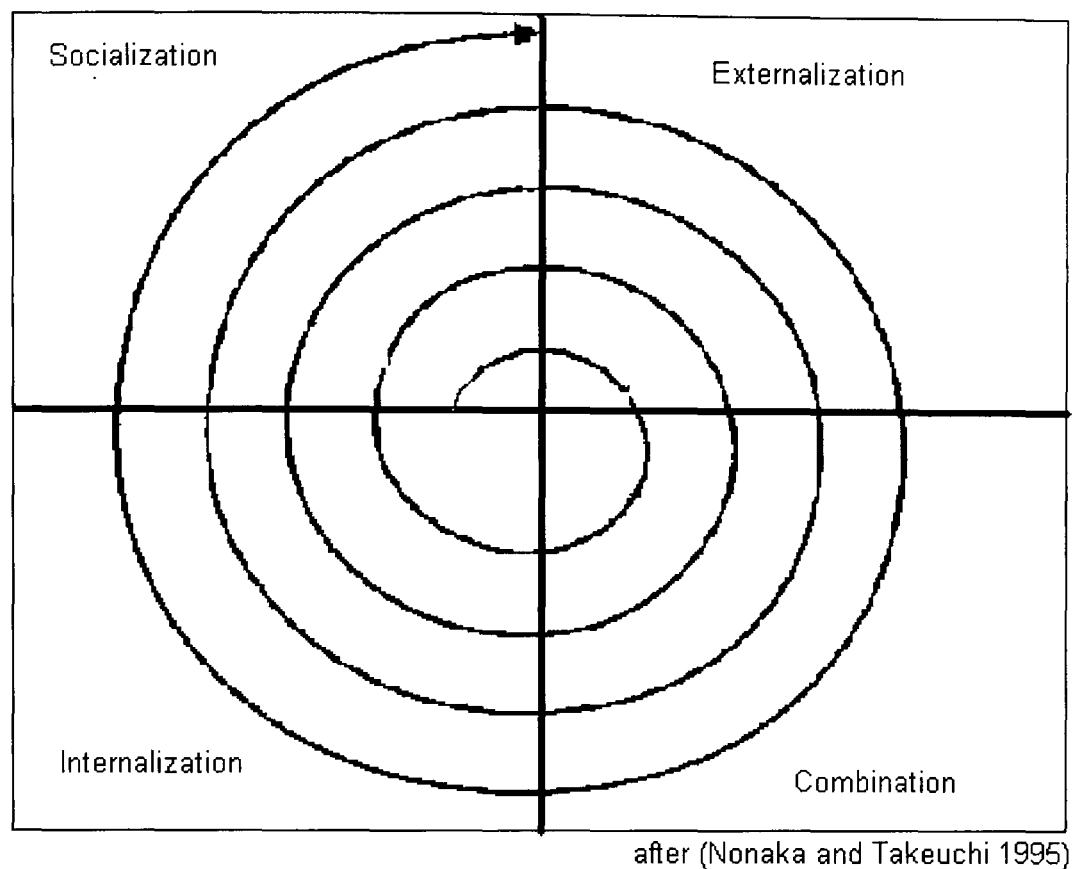
Knowledge creation refers to the development of ‘new’ organisational know-how and capability, Nonaka (1994) and Nonaka and Nishiguchi (2001). Nonaka and Takeuchi (1995) emphatically state that “knowledge is created only by individuals. An organisation cannot create knowledge without individuals.” They consider organisational knowledge creation as a process that “organisationally amplifies the knowledge created by individuals and crystallises it as a part of the knowledge network of the organisation.” Alavi (2000) presents the view that knowledge originates within individuals and social systems or groups of individuals. Alavi and Tiwana (2003) further suggest that at the individual level, knowledge is created through cognitive processes such as reflection and learning, while social systems generate knowledge through collaborative interactions and joint problem solving.

Learning is considered an important aspect of creating new knowledge. Vera and Crossan (2003) propose that “learning is the process through which knowledge is created and developed and, current knowledge impacts future learning.” Their observation is based on the premise that learning and knowledge are intertwined in an iterative mutually reinforcing process which presupposes that learning produces new knowledge which impacts future learning. Kolb (1984) identified four stages of a learning cycle which are concrete experience, reflective observation, abstract conceptualisation and active experimentation. Feedback and reflection play an important role in the progression of the learning cycle from concrete experience through active experimentation. Piaget (1970) is

also of the similar view while describing the learning process as dialectic between assimilating experience into concepts and accommodating concepts to experience. Dyba (2003) further extends feedback, reflection and a dialectic process that integrates local experience and organisational concepts into a dynamic model of software engineering knowledge creation.

Nonaka and Takeuchi (1995) draw upon the different perspectives of the Western approach and the Japanese approach to knowledge in their model of knowledge creation, Figure 1. They credit the Western approach of emphasising the management of explicit knowledge and present the Japanese approach of mobilisation and conversion of tacit knowledge as being the key to knowledge creation. Nonaka and Takeuchi view organisational knowledge creation as the interaction of tacit and explicit knowledge in the four modes of socialisation, externalisation, combination and internalisation. They call this interaction knowledge conversion and distinguish the four modes of knowledge conversion as:

- socialisation: conversion of tacit knowledge to tacit knowledge through social interaction
- externalisation: conversion of tacit knowledge to explicit knowledge through the use of metaphors and analogies
- combination: conversion of explicit knowledge to higher or different forms of explicit knowledge by reclassifying, categorisation and coding, and
- internalisation: conversion of explicit knowledge into tacit knowledge by doing, reading, discussion and practice



**Figure 1 Knowledge Creation Spiral**

Nonaka and Takeuchi (1995) also state that in the organisational knowledge creation process, the interaction between tacit and explicit knowledge and the consequent knowledge conversion emerges in a spiral that grows as the knowledge creation process progresses from the individual level to the group and organisational level. They list intention, autonomy, fluctuation and creative chaos, redundancy and requisite variety as conditions required at the organisational level to promote the knowledge spiral.

Nonaka and Konno (1998) suggest establishing an organisational 'ba' which they define as a common place or space for creating knowledge. They identify four types of 'ba' that correspond to the four modes on knowledge creation as:

- originating ba corresponding to the socialisation mode that facilitates social interaction and the transfer of tacit knowledge
- interacting ba which is associated with the externalisation mode and promotes dialogue and collaboration to convert tacit into explicit knowledge
- cyber ba which refers to the virtual space of interaction and corresponds with the combination mode, and
- exercising ba that facilitates the conversion of explicit to tacit knowledge in the internalisation process

'Ba' is similar to Demsetz's (1991) concept of common knowledge, or knowledge redundancy according to Nonaka and Takeuchi (1995), which refers to the common understanding of a subject area by organisational members who engage in communication.

Von Krogh et al (2000) further suggest sharing tacit knowledge, creating a concept, justifying the concept, developing the concept, and cross-levelling of knowledge as steps of knowledge creation. To facilitate knowledge creation they list instilling a knowledge vision, managing conversations, mobilising knowledge activists, creating the right context, and globalising local knowledge as knowledge enablers. The implied emphasis is also on interaction to share tacit knowledge and manage organisational relationships. This emphasis relates to Alavi and Tiwana's (2003) observation that groups, and therefore organisations, generate knowledge through collaborative interactions. Alavi and Tiwana further review how information technology supports knowledge creation.

#### **2.4.1.1 Information Technology Support of Knowledge Creation**

The primary role of information technology in knowledge creation is to facilitate interactions among individuals. In doing so, information technology helps support organisational collaboration, where collaboration denotes communicating and working together across organisational boundaries, (Baker 1992, Karsten 1999). Collaboration helps in combining and amplifying an individual's knowledge, thereby creating new knowledge at the group or organisational level. This knowledge creation is achieved through collaborative information technologies which aim to improve group collaborative interactions by providing techniques for structuring task interactions and systematically directing the pattern, timing, content, and recall of group discussions. Such interactions expose individuals to each other's thoughts, opinions, and beliefs while also obtaining and providing feedback from others for clarification and comprehension. Collaborative information technologies facilitate communication between individuals in an asynchronous and synchronous manner thus overcoming the barriers of time and space. Examples of such collaborative technologies include email, tele/video-conferencing, data-conferencing, web-based tools, intranets, e-learning systems, groupware technologies, electronic meeting systems, including bulletin boards and whiteboards. These technologies provide opportunities for extended interaction among organisational members for sharing their ideas and perspectives, and establishing dialogue, which may enable them to arrive at new ideas and interpretations, thus enabling new knowledge creation.

## **2.4.2 Knowledge Storage/Retrieval**

Effective knowledge management requires the storage and retrieval of organisational knowledge. Knowledge storage and retrieval refers to the development of organisational memory, Stein and Zwass (1995) and Walsh and Ungson (1991), and the means for accessing its content. Organisational memory includes knowledge residing in documents, databases, expert systems, procedures, processes, systems, and knowledge acquired by individuals. Walsh and Ungson (1991) refer to individual skills and organisational culture as internal organisational memory, which consists of the collective knowledge held by individuals or groups of individuals. Consequently, external organisational memory is considered to contain codified and explicit organisational knowledge and includes formal policies and procedure, and documents. Also, Stein and Zwass (1995) classify organisational memory as semantic or episodic. Semantic memory refers to general, explicit and articulated knowledge, while episodic memory refers to context-specific and situated knowledge.

### **2.4.2.1 Information Technology Support of Knowledge Storage and Retrieval**

Developing organisational memory requires determining the knowledge content of the memory and its sources, and specifying the means of accessing the content, Alavi and Tiwana (2003). Information technology is an enabler of the storage and retrieval of the semantic organisational memory. The technologies that enable the process are data warehousing, data mining and repositories. A data warehouse is a centralised repository that integrates, summarises and creates a historical profile of data which would otherwise remain fragmented, Inmon (1996). Data warehouses help convert large volumes of raw data into smaller chunks of interlinked information. The assumption underlying data warehousing is that valuable information is embedded in large volumes of objective data, Fayyad and Uthurusamy (1996). Data mining is the technique for uncovering such information and is defined as the process of automatically searching for unknown correlations in the data by looking for interesting patterns, anomalies and clusters, Loeb et al (1998). In most organisations, codified knowledge is often fragmented in many databases, Zack (1999). Repositories bring together content from various data sources thereby providing a unified access point and reducing knowledge search costs, Hansen (1999). Repositories can store highly structured content such as transactional data, customer records, and financial information, or relatively unstructured content such as



multimedia content and conversational discussion threads, Alavi and Tiwana (2003). Query languages, databases including multimedia databases, groupware technology and document management technology are also technologies that help develop organisational memory storage and retrieval.

### **2.4.3 Knowledge Transfer**

Knowledge transfer is a process that involves the transmission of knowledge from the initial location to where it is needed and is applied. It occurs at various levels within an organisation, and between individuals, from individuals to groups, between groups, across groups, between individuals and knowledge repositories, and between knowledge repositories. Communication processes and information flows drive knowledge transfer within organisations. Gupta and Govindarajan (2000) list the following as the elements of knowledge transfer:

- perceived value of the source unit's knowledge
- motivational disposition of the source
- existence and richness of transmission channels
- motivational disposition of the receiving unit, and
- absorptive capacity of the receiving unit

The last element, absorptive capacity, is defined by Cohen and Levinthal (1990) as the ability to recognise the value of new external knowledge, assimilate it, and apply it to commercial ends. The definition emphasises the three capabilities of recognising the value, assimilating, and applying new external knowledge, but also importantly implies the ability to utilise the knowledge made available through transfer. Further, knowledge transfer channels can be formal, informal, personal or impersonal, Holtham and Courtney (1998). Formal channels include training sessions, seminars and scheduled meetings, while informal channels include unscheduled meetings and discussions. Personal channels include apprenticeships and mentoring, while the more impersonal channels include repositories.

### **2.4.3.1 Information Technology Support of Knowledge Transfer**

Information technology helps ensure the flow of knowledge within an organisation. Information technologies that support knowledge transfer include discussion databases, knowledge maps, corporate directories, intelligent agent software, video conferencing, data conferencing, and electronic bulletin-boards and whiteboards. Alavi and Tiwana (2003) identify two models of knowledge transfer enabled by information technology: the network model and the stock model. The network model focuses on facilitating person-to-person transfer of knowledge via electronic communication channels, in synchronous and asynchronous manner. Alavi and Tiwana (2003) cite online chat, audio and video conferencing, email, voice mail and computer conferencing as technologies that support the network model. The stock model focuses on the electronic transfer of codified knowledge to and from computerised knowledge repositories, and the technology that best supports this model is the enterprise information portal. An enterprise information portal enables the transfer of knowledge from knowledge repositories to and from individuals through a central access point and a web browser interface. Alavi and Tiwana (2003) suggest the key advantage of enterprise portals as their ability to transfer knowledge to and from a diverse array of knowledge resources and backbone information systems from any place and at any time.

### **2.4.4 Knowledge Application**

Knowledge application refers to the use of knowledge for decision-making and problem-solving by individuals and groups in organisations. Knowledge requires its effective application to be a source of competitive advantage. Grant (1996) identifies three mechanisms for the application of organisational knowledge which are:

- Directives, which refer to the specific set of rules, standards, procedures, and instructions developed through the conversion of specialists' tacit knowledge to explicit and integrated knowledge for efficient communication to non-specialists, as proposed by Demsetz (1991).
- Organisational routines, which refer to the development of task performance and coordination patterns, interaction protocols, and process specifications that allow individuals to apply and integrate their specialised knowledge without the need to articulate and communicate what they know to others, and

- Self-contained task teams, which are formed of individuals with prerequisite knowledge and are effective in situations in which task uncertainty and complexity prevent the specification of directives and organisational routines.

#### **2.4.4.1 Information Technology Support of Knowledge Application**

Information technology supports knowledge application by embedding knowledge into organisational routines. This is made possible by information technology systems such as expert systems and decision support systems which are designed to apply the required knowledge for organisational routines. Expert systems include rule-based expert systems and case-based reasoning systems. Rule-based expert systems facilitate routine application of knowledge through codification of decision rules. Existing expertise is embedded in the software and use of the embedded knowledge is automated, George and Tyran (1993), Lado and Zhang (1998). Rule-based expert systems attempt to convert an expert's tacit knowledge into codified rules. Tiwana (1999) cites five conditions that must be simultaneously satisfied in order to achieve the development of a rule-based system for the purpose of knowledge application:

- the variables relevant to a problem should be known
- these variables should be expressed in quantitative terms
- the rules covering most if not all of these variables should exist
- rules that are applied simultaneously to solve a decision problem should not overlap
- the rules must have been validated in practice

Case-based expert systems facilitate the application of semi-structured knowledge through the intelligent reuse of previous problem-solving experiences. A case-based reasoning system documents each problem-solving initiative as a separate case, and over time, a collection of problem cases and solutions to the problems emerge. The system uses these cases, and their contextual details, to aid in solving new problems through a process of pattern matching, as cases that are the close matches are retrieved from the case base and further refined to find the closest match.

Decision support systems are defined as computer-based systems that support unstructured decision-making in organisations through direct interactions with data and analytical models McNurlin and Sprague (2001). Alavi and Tiwana (2003) cite the ability of decision-support systems to combine highly structured information with unstructured

information in a problem-specific context as their primary strength, and define dialogue, data and models as three components that define decision support systems. Dialogue represents the interactions between the user and the system, data represents aggregated data such as that obtained from a data warehouse, and models represent the analytical conceptualisations by which data are manipulated to address tasks.

Information technology also enhances knowledge integration and application by facilitating the capture, updating and accessibility of organisational directives by making them available on the organisation's intranet. This enables the knowledge to be available for speedier dissemination across organisational groups, departmental boundaries, time and space.

#### **2.4.5 Summary of the Knowledge Management Processes**

Information technology supports the four knowledge processes of creation, storage/retrieval, transfer, and application, which Alavi and Leidner (2001) consider are essential for effective knowledge management. Consistent with Davenport and Prusak (1998), and Zack (1999), Alavi and Leidner contend 'that the application of information technologies can create an infrastructure and environment that contribute to organisational knowledge management by actualising, supporting, augmenting, and reinforcing knowledge processes at a deep level through enhancing their underlying dynamics, scope, timing and overall synergy.' The information technologies and knowledge processes are not mutually exclusive, as a combination of information technologies maybe required, and the knowledge processes may not be in a linear sequence. Alavi and Leidner (2001) state that at any point in time and in any part of a given organisation, individuals and groups maybe engaged in several different aspects and processes of knowledge management, and thus emphasise that 'knowledge management is not a discrete, independent, and monolithic organisational phenomenon.' Several other studies concur with this observation, and with Alavi and Leidner's classification of the knowledge management processes. Grant (1996) states that effective knowledge management processes should be conducted frequently, consistently and flexibly. Also consistent with Alavi and Leidner (2001), DeLong (1997) classifies knowledge management processes into capturing, transfer and use of knowledge, while Leonard-Barton (1995) categorises the processes as acquisition, collaboration, integration and experiment. Knowledge management is therefore a broad organisational

concept which consists of a dynamic and continuous set of processes and practices embedded within the organisational individuals, groups and structures.

## **2.5 Knowledge Management: Strategies and Frameworks**

The section reviews the various knowledge management strategies that attempt to provide the framework within which organisations are able to manage their knowledge management initiatives. Sunassee and Sewry (2002) define a knowledge management strategy as a ‘high-level plan that aims at supplying the organisation with the knowledge resources that it needs to carry out its vision and goals,’ while Zack (1999) states that a knowledge management strategy expresses the overall approach an organisation intends to take to align its knowledge resources and capabilities to the intellectual requirements of its strategies.

In their framework for analysing knowledge management strategies, Rubenstein-Montano et al (2000) argue that most approaches to knowledge management do not adequately satisfy the knowledge management needs of organisations, and that there is a lack of cohesiveness across the various approaches. They recommend that all knowledge management approaches submit to the systems thinking method which provides the ability to view complex processes, a view similar to Senge’s (1990) regarding learning organisations. Rubenstein-Montano et al state that ‘people; the knowledge that people have, share and need; the culture for knowledge sharing (or lack thereof); organisational business strategies; and the technological infrastructure for knowledge management must all be considered for effective knowledge management initiatives.’ Systems thinking, they argue, can enhance knowledge management through its ability to depict complex, dynamic processes and thus enhance understanding and the ability of knowledge management initiatives to respond to the needs of the organization.

Rubenstein-Montano et al (2000) also classify knowledge management frameworks into three categories: prescriptive, descriptive, and hybrid. Prescriptive approaches provide direction on the types of knowledge management procedures without providing specific details of how the procedures can or should be carried out. Descriptive approaches describe knowledge management and identify attributes of knowledge management that can influence the success or failure of the initiative. The hybrid category is a combination of the prescriptive and descriptive approaches. The primary finding in their analysis of the

frameworks reveals that the existing knowledge management frameworks are primarily prescriptive, and that they tend to be task-based and neglect other aspects of knowledge management. This finding is consistent with Vera and Crossan (2003) who observe a 'strong prescriptive element in knowledge management which is understood as "managed learning" and is assumed to have a positive impact on performance.' Easterby-Smith and Lyles (2003) further endorse the same view by stating that discussions regarding knowledge management 'adopt a technical approach aimed at creating ways of disseminating and leveraging knowledge in order to enhance organisational performance, and that the role and design of information technology is often central to such discussions.'

Rubenstein-Montano et al (2000) recommend that a knowledge management framework should:

- be both prescriptive and descriptive, that is a combination of the two approaches
- be consistent with systems thinking
- link knowledge management to organisational goals and strategies
- be planned before the knowledge management activities take place
- acknowledge the organisational culture, and the knowledge management practices must be compatible with the culture
- direct knowledge management through learning and feedback loops

Vera and Crossan (2003) address the last recommendation in their framework which integrates the descriptive streams of organisational learning and organisational knowledge with the insights of learning organisations and knowledge management. Organisational learning focuses on learning as a process of change, while organisational knowledge stresses knowledge as a resource that provides competitive advantage and studies the processes associated with its management. Thus organisational learning and knowledge are facilitative and present a descriptive view. On the other hand, Vera and Crossan (2003) consider learning organisations and knowledge management share prescriptive views of how organisations should effectively learn and manage knowledge.

Earl (2001) proposes a unique knowledge management strategy of a framework that presents a taxonomy of "schools" for knowledge management. The taxonomy considers focus, aim, unit, success factors and 'philosophy' as the attributes of each school. According to Earl the schools of knowledge management are: systems, cartographic, process, commercial, organisational or social, spatial, and strategic. Earl labels the first three schools as "technocratic" because he suggests that they are 'based on information

technologies, which largely support and, to different degrees, condition employees in their everyday tasks.’ The fourth school is labelled “economic” being the most commercial in orientation, while the remaining three schools are viewed as more behavioural, as they stimulate management to be more pro-active in creating and sharing knowledge as a resource.

Other frameworks that attempt to provide a hybrid approach to knowledge management include Sunassee and Sewry’s (2002) theoretical framework, and Shukla and Srinivasan’s (2002) knowledge management architecture. Sunassee and Sewry provide a framework that consists of three main interlinked components: knowledge management of the organisation, knowledge management of the people, and knowledge management of the infrastructure and processes. The emphasis in their model is on the importance of aligning the knowledge management strategy of the organisation to the overall business strategy of the organisation, while the focus is on the importance of the employees of the organisation, and their contribution towards a successful knowledge management effort. Shukla and Srinivasan (2002) view knowledge management implementation as involving three significant decisions about: knowledge architecture, systems and technology, and people. The knowledge architecture decisions include knowledge mapping, creation of databases, and classification of knowledge. The decisions about systems and technology include implementing structures and systems for dissemination, access and usage, and technology used for accumulation, review, administration and access control of the knowledge base generated. The decisions regarding people relate to motivating them to participate in knowledge management programs, and creating a culture of knowledge sharing. These frameworks combined with the popular works of Alavi and Leidner, Nonaka, Davenport and Prusak, and others provide a broad perspective of knowledge management in the organisational context. A sample of frameworks analysed by Rubenstein-Montano et al (2000) are listed in Table 2, attached in Appendix 1.

The various perspectives, strategies and frameworks relate to the generation, capturing, storage, retrieval, transfer, dissemination, application and reuse of knowledge. However, as analysed by Rubenstein-Montano et al (2000), most frameworks tend to be prescriptive and focus on the codification strategy and the content perspective of knowledge management. As such, the frameworks enable the effective management of the explicit knowledge within the organisation. Knowledge management frameworks require an emphasis on the personalisation strategy with a relational perspective which facilitates the use and application of tacit knowledge also. However, the implications of the term “tacit

knowledge” need to be considered to better understand the flow of knowledge and the emphasis required from such knowledge management frameworks.

### **2.5.1 Knowledge Flows**

Polanyi (1967) proposed the dichotomy between the explicit and tacit dimension of knowledge, which was highlighted by Nonaka and Takeuchi (1995) in the processes of organisational knowledge creation. Nonaka and Takeuchi (1995) emphatically state that the cornerstone of their theory is the distinction between tacit and explicit knowledge, and that “the key to knowledge creation lies in the mobilisation and conversion of tacit knowledge.” Tacit knowledge, according to them, includes cognitive and technical elements, where cognitive elements are based on mental-models and technical elements include know-how, crafts and skills. Li and Gao (2003) observe that Nonaka and Takeuchi’s model is based on case-studies of “certain Japanese manufacturing companies that more or less relates to assembly lines, it is necessary to be cautious when the model is extended for a broader application.” Li and Gao argue that embedded within the model are the factors, issues and idiosyncrasies of Japanese society and organisational culture, particularly with regard to assembly line manufacturing, and that it is important to consider this while extending the theory into diverse disciplines and contexts. They contend that Nonaka and Takeuchi did not make a distinction between “tacitness” and “implicitness,” where the former is elusive and subjective awareness of an individual that cannot be articulated in words, while the latter can be articulated but is not done because of specific reasons under certain settings. Polanyi’s (1967) discussion allows that all knowledge has tacit dimensions, and Leonard and Sensiper (1998) suggest that instead of separate constructs, tacit and explicit knowledge signify a continuum, in which tacit refers to the extreme end where knowledge cannot be articulated, implicit refers to the mid-range of the continuum where knowledge is currently not articulated but could be made so, and explicit is the other end of the continuum where knowledge is articulated.

Tsoukas (2001), however, presents a contrary view that tacit and explicit knowledge are not two ends of a continuum but two sides of the same coin, and that even the most explicit kind of knowledge is underlain by tacit knowledge. This view is consistent with Polanyi’s (1967) implication that the elimination of the personal, tacit dimension in essence destroys all objective knowledge, as it provides the perception and mental models that enable the understanding of the comprehensive whole of an entity. Tsoukas contends that tacit



knowledge cannot be “captured,” “translated” or “converted,” but only displayed and manifested in actions, and that new knowledge comes about not when the tacit becomes explicit, but when performance is punctuated in new ways through social interaction. Though Tsoukas maintains a contrary view on the relationship between tacit and explicit knowledge, he concurs that social interaction is the basis for ‘new’ knowledge creation. Desouza (2003) is also of the view that social interaction facilitates tacit knowledge exchange by encouraging face-to-face dialogue between individuals of an organisation. Desouza states that organisational members exchange ideas and share narratives in informal settings more readily, thereby building a shared understanding. The informal settings, he argues, are emergent and more conducive to facilitating tacit knowledge exchange, than deliberate, planned interaction settings such as cross-functional team meetings, product innovation camps and brainstorming camps. Desouza’s (2003) study of an information technology company that introduced a game room as part of its people-centric approach to facilitate tacit knowledge exchange reveals that after the game room experience, electronic sharing of knowledge increased within the individuals of the organisation. Desouza argues that the distinction between tacit and explicit is tricky as the knowledge shared was explicit in nature such as references to manuals and web sites, or tacit knowledge “made” explicit such as suggestions on problem-solving methodologies, sharing personal experiences and opinions. The term “made” indicates Desouza’s emphasis of Nonaka and Takeuchi’s (1995) model and view that tacit and explicit knowledge are not totally separate but mutually complementary entities that interact with and interchange into each other in the creative activities of individuals.

Desouza (2003) studied the introduction of an electronic repository after the game room experience in the information technology organisation, and states that “in a twenty week time frame, the amount of tacit knowledge exchanged increased by 32%.” Multiple coders were used to carry out content analysis of the postings in the electronic repository and classify them according to various dimensions of tacit to explicit. Busch and Richards (2000) develop Formal Concept Analysis as a means of measuring articulable tacit knowledge based on the Likert scale approach proposed by Sternberg et al (1995). Busch and Richards (2000) further attempt to graphically define articulable tacit knowledge by using network maps to provide a visual perspective to what constitutes tacit knowledge. Busch et al (2001) classify articulable tacit knowledge and inarticulable tacit knowledge while using Social Network Analysis as a means of measuring the tacit knowledge diffusion within an organisational domain, by tracking the tacit knowledge flows within the information technology domain. Social Network Analysis essentially maps the

relationships between individuals and is adopted by Busch et al because of the social nature of tacit knowledge and its exchange.

The above discussion suggests that knowledge is viewed as tacit and explicit, and their various dimensions. In attempting to relate the two, researchers attempt to view and analyse the relationships between tacit, implicit and explicit, and articulable and inarticulable knowledge. Knowledge is viewed as what can be codified and stored in repositories, or as Polanyi (1967) suggests, what individuals know but cannot tell. Styhre (2003) views knowledge as what emerges in the notion of knowing within a “processual perspective of knowledge that conceives of knowledge as both what is manifested in practices and simultaneously endowed within a conceptual framework.” Styhre (2003) states that knowledge exists throughout an organisation and is not a clearly bounded and manageable resource that can be located in one single point in time and space. In other words, knowledge is fluid and emergent, and not fixed and stable, and being fluid and moving, it is embedded in social relationships, and emerges in practices and the use of concepts. Further, Styhre (2003) argues that decision-making, creativity and innovation may take place anywhere within an organisation rather than locating the various qualities and practices in specific locations, and therefore knowledge is required to exist within the processes of an organisation.

### **2.5.2 Knowledge Sharing**

The knowledge that exists within the processes of an organisation can be utilised in knowledge sharing activities which Grant (1996) states “must combine the explicit information held in documents with the specialised tacit knowledge possessed by individuals.” According to Grant this involves generic activities such as:

- transferring experience and organisational learning
- keeping track of process knowledge and using it to find the best plan to achieve stated goals
- building collaborative knowledge, where individuals with diverse expertise combine their knowledge to create new products and services
- developing transactional knowledge, which relates to internal and external knowledge

Hawryszkiewicz (1999) defines knowledge sharing processes as being composed of generic activities that are combined into processes and adapted to a particular need. According to Hawryszkiewicz the activities include:

- establish common ground
- quickly adapt to different terminologies in distributed systems in order to make sense of available documents and information
- form shared perspectives, even when different terminologies are used
- interpret and transfer knowledge

Hawryszkiewicz (1999) observes that information technology has to date supported prescriptive processes, but its support for knowledge sharing must go beyond prescriptive processes and provide the ability for processes to dynamically emerge as individuals' understanding of a particular situation grows. Castells (1996) states that information technology supports the "pervasive expansion of networking throughout the social structure, and that CSCW-solutions provide a good illustration of this, with the promise of standardised approaches to knowledge sharing in organisations." CSCW is Computer Supported Cooperative Work which involves collaborative information technology that supports communication and collaboration between individuals.

Walsham (2001) cautions against a standardised approach to information technology knowledge sharing. Walsham's (2001) research findings indicate a great variance from the reported success from applying groupware technologies and other CSCW-solutions to support inter-functional knowledge sharing and collaboration amongst groups. Ciborra (1996) states that positive outcomes from groupware applications depend strongly on the match between the technology and the practices of the individuals involved. Further, mere adoption of the collaborative information technologies does not achieve its intended purposes, and widespread deployment does not necessarily follow widespread acquisition of collaborative information technologies by acquiring organisations, Fichman (1999). For collaborative information technologies to have a positive impact on quality, productivity and collaboration, they must be accepted and used by employees in organisations. The success of collaborative information technologies depends upon how individuals use them which in turn effects how knowledge is shared, Constant et al (1996). Development of individuals' motivation and ability to communicate and share knowledge and experiences are required for organisations to benefit from the knowledge shared amongst its individuals.

## 2.6 Conclusion

This chapter presents the organisational context of knowledge. It reviews the knowledge management processes and the role played by technology in supporting them to share and utilise the knowledge generated. Information technology support often results in a greater emphasis on managing explicit knowledge within knowledge management initiatives. Different knowledge management strategies that attempt to provide the framework within which organisations are able to manage their knowledge management initiatives are discussed. The different perspectives on tacit and explicit knowledge are highlighted, as is the importance of identifying the knowledge flows within an organisation. Though information technology is an integral part of the knowledge management infrastructure and plays an important role in facilitating knowledge sharing, the social issues regarding knowledge sharing are crucial. Knowledge is viewed as a set of shared beliefs that are constructed through social interactions amidst certain social circumstances, Berger and Luckmann (1966). Also, knowledge is fluid and moving, embedded in social relationships and emerges in the practices and use of concepts, Styhre (2003). Organisations need to address the social issues to ensure their knowledge management initiatives are effective.

However not all issues, social or technological, are resolved within clear organisational boundaries and structures and some require more urgent resolution forms. Compared with the more permanent structure of organisations, projects provide temporary mechanisms for facilitating change and achieving specific objectives in the short term. Under project conditions the issues related to knowledge management become more critical given the need to ensure continued success. The next chapter discusses the use of knowledge under such project conditions.

## Chapter 3 Knowledge within Projects

The previous chapter focused on knowledge structures within organisations. This chapter discusses the relationship between projects and knowledge, and in particular, how the use of knowledge within projects differs from utilisation of knowledge within organisations. The chapter also explores the different types of knowledge used in projects, and how project teams function as compared to work structures in organisations and communities of practice.

### 3.1 What are Projects?

The previous chapter discussed the use of knowledge within organisations which are viewed as collective groups devised and maintained to achieve specific objectives. Most literature on knowledge processes relates to organisations that are considered to be permanent, long-term structures with routines, policies, systems and procedures, Nonaka (1994), Nonaka and Takeuchi (1995), Von Grogh (1998), Von Grogh et al (2000), Earl (2001), Sunassee and Sewry (2002), Shukla and Srinivasan (2002), and Vera and Crossan (2003). Morgan (1997) views organisations as machines, organisms, cultures, and political systems and also states that they have an ability to learn. Organisations consist of structures that perform functions to achieve tasks, goals, aims and objectives, and they attempt to perform these functions in their day-to-day activities. They can also be viewed as a system of coordinated behaviour including rules, regulations, procedures and standards, (Mitchell and Larsson 1987). According to Kast and Rozenzweig (1985) organisations rely on planned structures that represent a deliberate attempt to establish patterned relationships among components. The structure of organisations is intimately linked to functions and is set forth by the design of major components or subsystems and the establishment of patterns of relationships between them.

Earl's (2001) taxonomy of schools of knowledge management depicts the various approaches available to organisations in developing and managing their knowledge initiatives. While the use of information technology is often favoured, organisations also lay an emphasis on the social, spatial and strategic aspects, (see Section 2.5). Koskinen (2003) describes organisations as physical structures, which Davis (1984) states have "the architectural design and physical placement of furnishing in a building that influence or

regulate social interaction.” Davis (1984) and Earl (2001) suggest that physical structure can influence the type of interaction that occurs within and among individuals in an organisation. According to Peters (1990), individuals most likely interact with others when the physical characteristics of the building or settings encourage them to do so. Also, organisations exist to achieve long term objectives and goals. As a result they aim to establish a knowledge management initiative as a permanent process in their systems and operations, and thereby attempt to capture knowledge in routine and repeatable situations.

Projects, on the other hand, are designed to achieve specific objectives within a predetermined time frame, budget and resources. Projects involve planning for non-routine tasks to be conducted in several phases, and can be characterised as unique, goal-oriented and complex undertakings steeped in uncertainty, which aim to produce a meaningful product or service in order to satisfy a need, Dalcher (2003a). Davis (1951) defines a project as “any undertaking that has definite final objectives representing specified values to be used in the satisfaction of some need or desire.” The Project Management Institute (PMI) makes a distinction between ordinary work and projects by emphasising the temporary and unique nature of projects (PMBOK 2000).

The key characteristics that differentiate projects from organisations are:

- Projects start with a definable purpose
- Projects end with a final deliverable or result
- Projects represent a unique undertaking
- Projects are temporary activities, with a definite and limited time span
- Projects are about change and are uncertain
- Projects are an effort where there is something at stake
- Projects are processes of working towards a goal

### **3.2 Project Management**

Projects generate productivity, efficiency and effectiveness by virtue of horizontal work flow, Kerzner (2004). Horizontal management organises work across various functional groups resulting in improved coordination and communication among employees and managers and therefore enabling more work to be accomplished in less time by less people. Project management is expected to result in better customer interaction, decision making, problem solving and quality, Kerzner (2004). In order to achieve these results

Kerzner (2003) defines project management as the ‘planning, organising, directing, and controlling of company resources for a relatively short-term objective that has been established to complete specific goals and objectives. Furthermore, project management utilises the systems approach to management by having functional personnel, the vertical hierarchy, assigned to a specific project, the horizontal hierarchy.’

The Project Management Institute (PMI) defines 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.” Therefore, the PMI views project management as an organisational means of achieving strategic goals by implementing multiple projects that can be grouped within portfolios or programmes. The PMI’s perspective views a group of projects as a program, and also, a group of project or programs as a portfolio. A contrary view is often adopted within industry where a group of projects are considered a portfolio, and a group of portfolios are considered a program. Within both views, project management helps implement business strategy, and provides a high level perspective and regulation of critical resources that directly impact organisational performance.

### **3.3 Project Management Maturity**

As organisations implement multiple projects within their portfolios and programmes, they require consistent, reliable, repeatable and standardised processes to execute their projects. Though projects are unique to a relative degree, they possess some common characteristics and processes within an organisational perspective, which can be improved and made more effective. Effective project management requires extensive planning and coordination supported by a project management methodology that provides a repetitive process that can be used on subsequent projects. Kerzner (2003) provides the following definition:

“Maturity in project management is the implementation of a standard methodology and accompanying processes such that there exists a high likelihood of repeated success.”

Maturity is considered to be a measure of standardisation, consistence, process conformity and organisational behaviours. The definition implies that a proper foundation of tools, techniques, processes and culture are required to attain project management maturity. This repeatable and optimised foundation is expected to provide predictable, reliable and

consistent project outcomes. A mature project organisation constantly aims to improve its processes, and is therefore more proactive than an organisation that has not attained project management maturity.

### **3.4 Knowledge use in Projects**

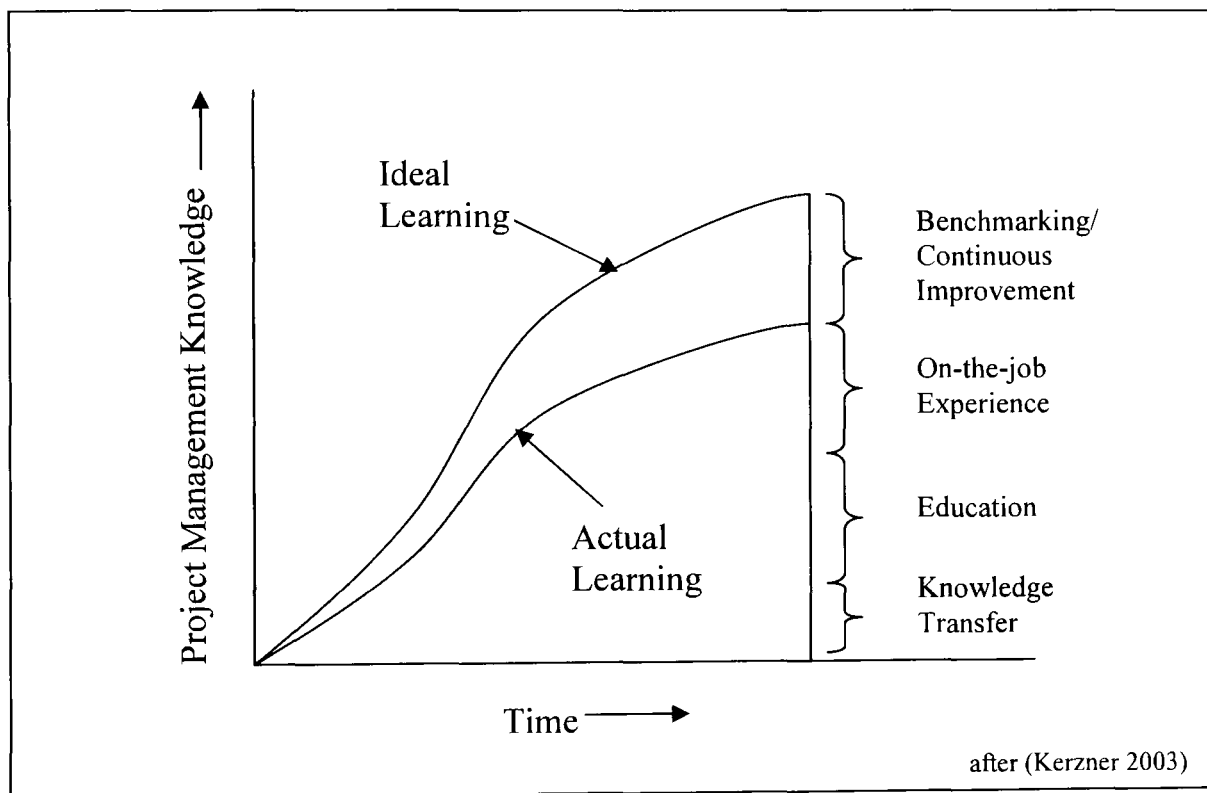
The use of knowledge helps projects achieve the required project outcomes. Projects utilise and leverage knowledge that was either developed internally or acquired elsewhere. The development and implementation of projects depends to a large extent on the individual knowledge of project personnel that include stakeholders, clients, and users, and their access to knowledge resources, and further integration of knowledge generated within the project. Projects therefore produce valuable knowledge that can be shared, and enhance existing knowledge in other developmental projects. Stewart (1997) states that 'projects package and sell knowledge' and therefore help 'create new value.'

The temporary nature of projects, and their time constraint, makes the capture of knowledge unique. In contrast to projects, organisations establish knowledge management programs as permanent processes in their operations, while in projects the opportunity to capture knowledge is limited by time. Projects require that critical knowledge is identified and captured for utilisation. This knowledge is then combined with individual and collective team competencies for effective reuse. Members of project teams possess diverse skills, capabilities and competencies developed through experience. However, as projects are temporally limited, the lessons learned and knowledge created within the teams are dispersed when the project ends. Systematic and efficient knowledge management requires a long term perspective of project management that supports a commitment at the outset of a project to capture the learning and knowledge generated by the project. Most management literature emphasises the key role played by past experience in improving management performance. Effective leveraging of past experience depends upon how well and systematically this experience is captured and organised to enable learning and reuse. Systematically recording data from projects, deriving lessons learnt from them, and then making the lessons available to other projects, enhances project capabilities through reuse. Koskinen (2003), while evaluating knowledge within projects, considers tacit knowledge to represent knowledge based on the experience of individuals and cites memory systems, communication systems, motivational systems and situational systems as groups of factors that affect its utilisation within projects. Terming situational

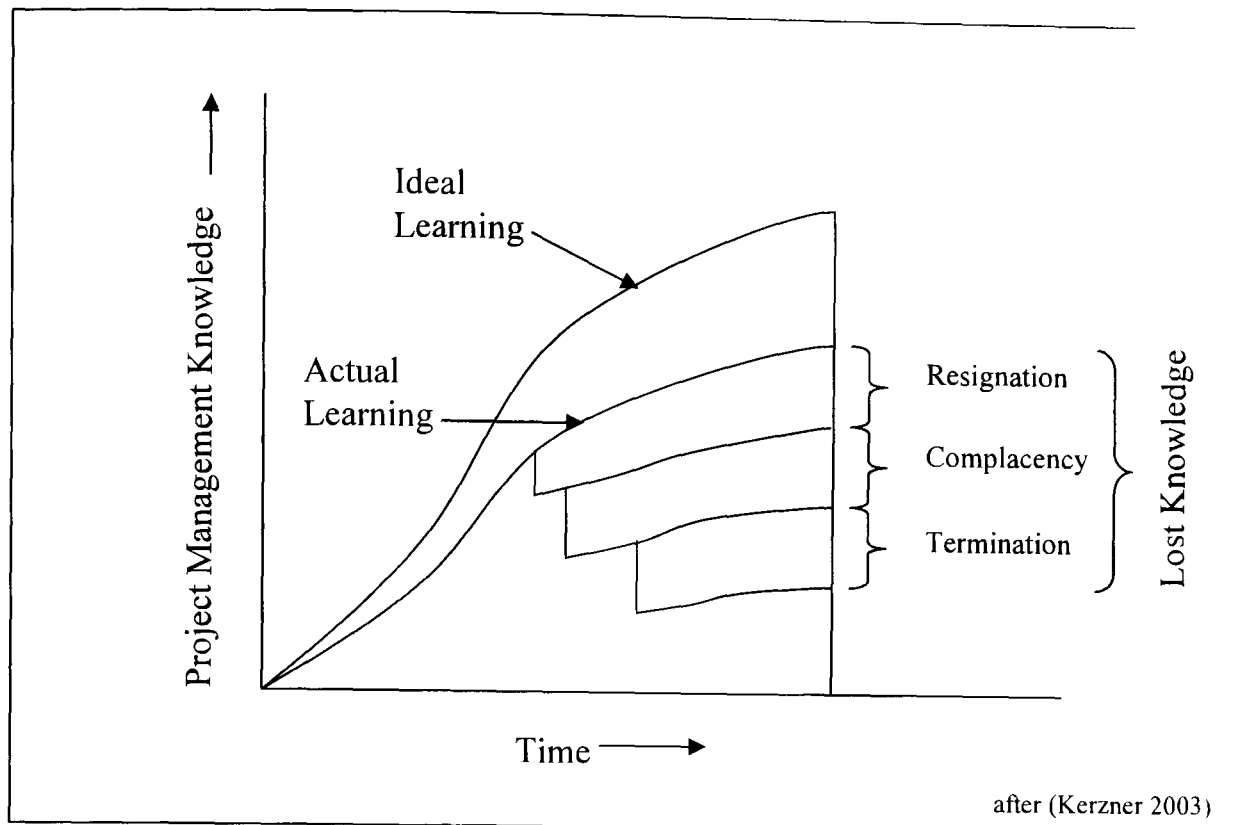


systems to be an external factor, Koskinen suggests that it forms the background within which the internal factors of memory systems, motivational systems and communication systems form a three-dimensional space which helps to estimate the role of tacit knowledge and its utilisation within a project organisation.

Kerzner (2003) states that knowledge transfer, on-the-job experience and education result in actual learning within a project environment. Kerzner contrasts actual learning with ideal learning, which he states is obtained through lessons learned, benchmarking and continuous improvement efforts, Figure 2. However Kerzner (2003) also asserts that while ideal learning is 'hardly ever reached', actual learning is less than expected due to lost knowledge as depicted in Figure 3.



**Figure 2 Project Management Knowledge**



**Figure 3 Lost Project Knowledge**

Further, Bredillet (2004) states that for the past forty years projects and their management have become a well-accepted and strategic way to manage organisations. According to Bredillet (2004), projects as strategic processes modify the conditions of the organisation and its environment. Meredith and Mantel (1995) describe three basic types of project organisations:

- projects as a functional section of a larger organisation
- pure project organisations where a project is accommodated in a separate, largely self-contained section that is disbanded when the project is completed or disbanded
- organisations where projects are run on matrix basis in which control rests with a project manager but the majority of human and other resources are borrowed from different sections of the larger organisation

Knowledge is required to implement the project in relation to the type of project organisation. Koskinen (2004) suggests a metaphor of a project tree to visualise the entire knowledge required by a project organisation and states that the types of knowledge that a project may require are tacit, explicit, additive or substitutive. Koskinen refers to additive and substitutive knowledge as knowledge that is new to the project and is either invented internally or acquired from external sources. This is similar to Bredillet's (2004) view that project teams need to know what knowledge is available to complete the project based on past experience, and what knowledge needs to be acquired or will emerge as a result of the

unique nature of the project tasks. Leseure and Brookes (2004) distinguish between kernel and ephemeral knowledge by labelling kernel knowledge as knowledge similar to the core competencies of an organisation and includes forms of knowledge that need to remain and be nurtured within an organisation in order to sustain high project performance in the long-term, and ephemeral knowledge as project-specific knowledge that is active and useful only during the lifetime of a project. Following the identification of the knowledge required, Koskinen suggests four different environments that help understand situations where knowledge management processes take place in the project management context. The four different environments, consistent with Koskinen's (2003) situational systems, are: mechanical project management environment, semi-mechanical project management environment, organic project management environment, and semi-organic project management environment. The projects are identified or classified in an environment and the type of knowledge required is assessed and applied.

In contrast to Koskinen's (2003, 2004) distinction or compartmentalisation of situational environments is the 'communities of practice' perspective. Wenger and Snyder (2000) define communities of practice as "groups of people informally bound together by shared experience and passion for joint enterprise." Communities of practice develop when there are ample opportunities for informal contact. Over time, the interactions among individuals enable them to develop their own routines and share assumptions, experiences and knowledge. Leonard and Sensiper (1998) state that a certain level of personal intimacy is necessary that allows the communication and transfer of tacit knowledge. Communities of practice evolve practices, and often create their own languages, which may contain jargon and colloquialisms whose meanings outsiders may not be familiar with. They may or may not be a part of a team structure as organisational teams are often hierarchical by nature, and project teams are unified by way of a specific goal and deliverable, while communities of practice are informal groups that share knowledge about a common topic. Bredillet (2004) provides a distinction between project teams, communities of practice (see Section 2.1) and 'ba' (see Section 2.4.1), and a perspective on how they learn and share knowledge, attached in Appendix 2.

Bredillet (2004) also states that competencies, which are work-related behaviours, are at the source of competitive advantage and the creation of value. Bredillet views competencies as a resource and "the most relevant driver," and states that through a project organisation's processes, past action is actualised as experience, present action reveals and proves competencies, and future action generates and tries out new competencies.

Koskinen (2003) divides an individual's competence into three parts as tacit knowledge, attitudes, motivation and other personal characteristics, and explicit knowledge, implying that knowledge, tacit and explicit, is the underlying basis for forming competence. However, von Krogh and Roos (1996) state that "knowledge is about specific insights regarding a particular topic; competence is about the skill to carry out work." This implies that skills and capabilities of an individual are not fixed properties, but are produced in the individual's situated practices. Koskinen (2003) states that "when a performance of an individual is seen as his or her dynamic engagement to a task, the personal competence is understood as emerging from situated practices." The flow of knowledge within the knowledge processes of an organisation or project contributes to the individual competencies. The experienced individuals, or knowledge experts, consequently play a key role in the creation and diffusion of tacit or kernel knowledge as they are often unique sources of critical forms of knowledge within a project team or within the whole organisation (Leseure and Brookes 2004). Knowledge maps help locate such experts within project organisations and enable the formation of expert networks to establish knowledge sharing and improve decision making and problem solving. The tacit or kernel knowledge of individuals is required in development projects which lay an emphasis on the knowledge and creativity of the individual developers working in teams.

### **3.5 Project Teams**

Hovarth et al (1996) view a team as a socially constructed phenomenon or linking mechanism that integrates individuals and organisations. Project team members with diverse skills, knowledge and experiences are required to work together to resolve issues while developing new solutions. Innovation and product development have been described as knowledge intensive activities, Iansiti and MacCormack (1997). Developing new products or solutions entails the application of knowledge to new problem-oriented situations and often involves cross-functional linkages, where individuals join a team with differing viewpoints. The interaction within such team members brings about a need to organise, integrate, filter, condense and annotate the collaborative data and other relevant information that the team members contribute, (Morrison and Kennedy 1996), resulting in new ideas and solutions that did not previously exist within the organisation. Cross-functional project teams are used to generate consensus through collective input, investigation and negotiation, and also to manage strategic change initiatives. The diversity and strategic value of the team members' specialised knowledge enhances the

competitiveness of the project organisation. However, this competitiveness depends on the project organisation's capacity and ability to integrate the knowledge in an effective manner.

Fong (2005) identifies knowledge integration, knowledge sharing and collective project learning as three modes of knowledge creation within multidisciplinary project teams. Fong (2005) views knowledge sharing as 'a multitude of processes taking place directly without language (socialisation) and with language (externalisation)' and knowledge integration as 'the willingness to combine knowledge from within and outside the team.' Grant (1996) states that an organisations knowledge integration capacity is determined by two crucial mechanisms: direction and organisational routines. Direction enables communication between specialists by codifying tacit knowledge into explicit rules, Demsetz (1991), and organisational routines reduce the need for communicating the explicit knowledge within project teams. Collective project learning has an open-ended and long term focus where an important aspect is the willingness to combine knowledge from within and outside the team, (Fong 2005).

Project learning provides a challenge to establish how social practices facilitate or inhibit learning and therefore to find ways of reconfiguring them to enable more effective cross-project knowledge diffusion and learning. Precise project objectives and the finite lifespan of activity provide project teams with an immediate focus, which however, can work against the long term focus of sharing experiences, understanding and learning. During a project, effort is often focused on immediate deliverables with no emphasis on how to help future projects.

### **3.6 Conclusion**

The chapter presents the use of knowledge in projects and how it is different from the ongoing knowledge management initiatives established by organisations. It views how mature project organisations attempt to have a standard methodology to implement projects. The chapter discusses the different types of knowledge that projects can draw upon, and also distinguishes individuals working together in project teams as compared to the concept of communities of practice. During a project, effort is often focused on immediate deliverables with no emphasis on how the experience and insights gained would help and benefit future projects. Project-based organisations work on time-scales that are

typically organised around teams assembled specifically for the project and often disbanded upon its completion. Personnel often change during the course of projects and team members also often come together for the first time at the outset of the project therefore making it is difficult to create the right knowledge culture and effectively transfer knowledge and project learning to subsequent projects. The next chapter further refines the context by looking at knowledge in projects related to software development.

## **Chapter 4 Knowledge in Software Development Projects**

This chapter presents the need to use knowledge in software projects and the different practices adopted in an attempt to capture and disseminate this knowledge. The chapter also discusses the significance of the software development processes and how software organisations try to achieve repeatability and continuous improvement in their processes.

### **4.1 The Need for Knowledge in Software Projects**

The requirements of software projects are subject to frequent change even during later stages of the development process. Such change and uncertainty make software projects more unpredictable than other projects. Software projects are organised around teams and rely upon the knowledge, creativity and competence of the individual team members.

Myers (1985) states that more than half the cost of complex software development is attributable to decisions made in the ‘upstream’ portion of the development process, namely, requirements specification and design. Traditional software development models like the waterfall and spiral follow a sequential approach where the ‘upstream’ and ‘downstream’ aspects of development are clearly distinguishable. However, modern approaches like agile methods are more emergent and evolutionary, and rely on frequent feedback and interaction between and within self-organised teams while attempting to address change and uncertainty in the requirements. Self-organised teams comprise a group of peers who possess a sense of ownership and share responsibility for managing their own work, which includes problem-solving and continuous improvement of work processes. Gruenfeld et al (1996) suggest that trust is becoming increasingly important within such self-organised teams, as team members require it to respond openly and incorporate new knowledge in order to develop useful decisions. Politis (2003) also states that trust enables team members to create and share new knowledge, resulting in improved decision making and, furthering the ability of the teams to deliver the best possible solutions. Knowledge is the raw material required for decision making within software design teams and for complex projects, knowledge from multiple technical and functional domains is required. Curtis et al (1988). Walz et al (1993) state that ideally, software development teams are staffed so that both the levels and distribution of knowledge within them match those required for the successful completion of the project.

However Curtis et al (1988) concur with Koskinen (2004), Bredillet (2004), and Leseure and Brookes (2004), that knowledge shortfall, such as the thin spread of application domain knowledge, often occur. In general, individual team members do not have all the knowledge required for the project and must acquire certain additional knowledge for accomplishing productive work and completing the project. The sources of such additional or ephemeral knowledge (see Section 3.4) could be relevant documentation, formal training sessions, or the diverse knowledge of other team members. Knowledge is relevant and required in various aspects of the software development process, such as:

- estimation of costs and time scales
- communicating with clients and users
- problem solving and decision making
- design
- reuse of code
- training
- maintenance and support

The need for knowledge in software development projects extends beyond domain knowledge and decision making. Knowledge and reuse of previous work helps in avoiding mistakes, and reduces the time and cost of the developmental process. Brossler (1999) states that development teams do not take full advantage of existing experience and repeat mistakes over and over again. Repeating successful processes increases the productivity, quality and the likelihood of success as knowledge gained in previous projects helps in avoiding mistakes and repeating success for future projects. Knowledge is required to maintain continuity and avoid knowledge gaps in the event of change in the developmental team composition, as Brossler states that ‘if a person with critical knowledge leaves, severe knowledge gaps are created.’ Such a change in team composition requires the new team members to be acquainted with the team and organisational culture, policies, practices, and sources of knowledge. Knowledge is also required to be shared between different groups and teams during collaborative work which is increasingly being increasingly utilised in software project organisations.

## **4.2 Managing Knowledge in Software Projects**

Software organisations possess knowledge that is diverse and engage in various activities and strategies to leverage this knowledge. Document management, competence



management, collaboration, expert networks and software reuse are some of the strategies employed to leverage such knowledge. Lindvall and Rus (2003) identify configuration management, design rationale, traceability, defect tracking, CASE tools, prediction models, case-based systems, and lessons learned and best practices as tools that help create a memory, which can be leveraged in order to implement knowledge management strategies within software organisations. In her theory of knowledge reusability, Lynne-Markus (2001) describes the knowledge reuse process in terms of four stages: capturing or documenting knowledge, packaging knowledge for reuse, distributing or disseminating knowledge, and reusing knowledge. Aurum (2003) highlights the need to capture and share emerging knowledge in software organisations, and to support the systematic storage of evolving knowledge. This requires software organisations to capture, share, coordinate and manage the knowledge, while finding complete solutions to the problems at the project level, and to find and integrate partial solutions for continuous improvement. Subsequent projects teams would be able to draw upon the experiences, learning and solutions of past software projects. Henninger (1997) recommends the systematic institutionalising of this knowledge by having a repository of project experiences, including tools and methods. The benefits from such a repository include lessons learned, commonly occurring patterns and reduction of duplicated effort. Such knowledge when made available and used proficiently, improves the overall quality of the software developed.

Jalote (2003) presents a strategy whereby the lessons learnt from projects are captured as process assets, and stored in a process database and a system called the body of knowledge (BOK). This knowledge is then leveraged to improve the execution of future projects. Consistent with Styhre's (2003) view that knowledge exists within the processes of an organisation, Jalote states that the centrepiece of a knowledge infrastructure for project management is the processes and related process assets, as a process oriented approach for project execution forms the foundation of a knowledge management system. Processes describe how different tasks are to be executed and encapsulate the knowledge the organisation has for efficiently performing that task, and process assets are documents that aid in the use of processes, while a process database keeps a summary of past projects. Processes capture past experience in executing projects, and if repeated successfully, enable the organisation to leverage this experience in future projects. Essentially, processes incorporate what the organisation has learned about successfully executing projects, Jalote (2000). An established approach for improving the processes of a software organisation is to enhance the processes based on experience gained from successful and failed projects.

### 4.3 Software Project Organisation Maturity

The processes used for executing a software project influence the effect on quality of the software produced and the productivity achieved in the project. Software project organisations aspire to attain repeatable and predictable processes, and continuous improvement of those processes. The Capability Maturity Model Integration (CMMI) for software is a framework that addresses the need to evaluate and improve processes used in an organisation for executing software projects. A process framework specifies some characteristics that the process must have to ‘qualify’ as a process associated with a certain level of maturity. The CMMI outlines software development processes and activities and helps distinguish mature processes from ad-hoc processes. The framework classifies the maturity of the software processes in five levels, with level 5 being the highest maturity, similar to Crosby’s quality management maturity grid which describes five evolutionary stages in adopting quality practices in an organisation. Organisations with higher CMMI levels are expected to operate using a more stable and disciplined approach, making use of automated tools and the experience gained from past projects. Each maturity level from level 2 to level 5 is characterised by key process areas, which specify the areas on which the organisation should focus to elevate its processes to that maturity level. For continuous improvement of their processes, software organisations need to invest an effort to capture the experience of implementing successful and failed projects within its processes and subsequently leveraging this experience. Learning and leveraging experience with processes constitutes an important aspect of level 3 of the CMMI. At level 3 the software processes for the project organisation are defined and implemented, enabling the organisation to learn from different projects and subsequently improve the processes to benefit future projects.

The CMMI summarises the best practices for software development within a software project organisation. As understanding of the repeatable processes increases, the processes are standardised with the objective of avoiding certain mistakes and promoting productivity. A mature software project organisation possesses an organisation-wide ability for managing software development and maintenance processes. These should result in the development of quality software within schedule and cost based upon realistic estimates determined by historical performance. Mature organisations possess an objective, quantitative basis for judging product quality and analysing if any problems exist within the products or processes. Therefore, while the CMMI is an appraisal method, progressing

through the stages defined within its framework, is meant to enable a software organisation to develop its processes and improve product quality.

However, while the software processes are increasingly standardised, some software development situations require more flexibility than can be addressed by the standardised processes. Rubinstein and Pfeiffer (1980) state that the problem-solving ability of an organisation can be hindered by repeatedly attempting to solve a problem using an approach that has been performed successfully in the past. While such an approach may be effective at times, differing situations require different problem-solving approaches. Also, Simon (1977) distinguishes between structured and non-structured situations, where repetitive and routine structured situations are addressed by standardised processes and operating procedures, while unstructured situations need human judgement, insight and intuition to be resolved. Software projects are by definition novel, unique and different, (see Section 3.1) and present unstructured situations that require insights and tacit knowledge of individuals for creative problem-solving. Therefore, while the CMMI helps standardise software processes, identify best practices, and reduce mistakes and rework, it also reduces a project organisation's ability and flexibility to adopt different approaches to problem solving that may result in vital learning.

Levitt and March (1988), are of the view that organisational learning is the experiential production and reproduction of organisational rules leading to behavioural stability or behavioural changes. Further, March (1999) states that "as a result of their experiential learning, organisations generate competence in the form of either exploitation or exploration." Exploitation creates reliability in experience through refinement, routinisation, production and implementation of knowledge. Exploration, on the other hand, creates variety in experience through search, discovery, innovation and experimentation. Also, learning competence means that organisations become better at things they do repeatedly and successfully and that they become less competent at things they do infrequently and unsuccessfully, Holmquist (2004). Therefore in the context of software organisations, there is need to find a balance between exploiting previous successes and exploring new possibilities and solutions. While process improvement approaches like CMMI help standardise reliable processes, software organisations also require harnessing the experiential knowledge created to help the more effective and flexible implementation of future projects, which are by definition novel, unique and different, (see Section 3.1).

## 4.4 Conclusion

Software projects require knowledge to help effectively implement projects by decreasing time and cost and improving quality. Software development is considered to be a human-based knowledge intensive activity. The success of software projects depends upon the knowledge and experience of the individuals who work in software project teams. Software organisations therefore need to adopt a learning approach. As such, software organisations attempt to capture and leverage the experience gained while implementing projects within their development processes. Certain software process improvement approaches, for example CMMI, suggest that the development process be standardised to deliver the most of the software organisation's capability. A standardised approach can however, hinder a software organisation's problem-solving ability, especially when dealing with unique undertakings and complex projects. Software organisations need to adopt a long-term approach to allow continuous learning and feedback through on-going interaction to leverage the creativity and experiential knowledge of their teams in order to benefit future projects. Based upon this need of software organisations, the next chapter develops the research hypothesis, and discusses how adapting a published and accepted model within the literature helps in framing the hypothesis and research question.

## Part II    Research Methodology

## Chapter 5 Defining the Research Problem

Part 1 of the report presents the need for knowledge, existing and created, to benefit future projects within software project organisations. Learning is an integral part of knowledge activities such as knowledge creation and sharing, and an emphasis on learning and knowledge management creates intangible knowledge assets, which according to Wiig (1993) include “experience, expertise, proficiency, competency, skill, capabilities and embedded knowledge of all kinds.” This chapter explores how such experience, capabilities and skills can be leveraged from one project to subsequent projects by adopting a continuous and long term perspective that enables an organisation to benefit from knowledge assets. The research problem is defined and a published and accepted model that facilitates feedback is presented, and further extended and adapted to capture the essence of the research question.

### 5.1 Hypothesis and Research Question

The knowledge that a software organisation creates and the processes used for creating it are based on the organisation’s prior experiences of problem-solving, as discussed in the previous chapter. While software organisations identify best practices and lessons learnt and attempt to standardise them within their developmental processes, they need to adopt a flexible approach to problem-solving in dynamic situations, see Section 4.3. Such an approach is essential within teams of software developers working together to develop solutions they could not develop by working alone as individuals. As discussed in Sections 3.2 and 4.1, the focus on teams and their collaborative processes is significant as no individual embodies the breadth and depth of knowledge necessary to comprehend complex software systems and their solutions. Therefore, while individuals work together and their interactions help create knowledge, the learning from implementing a software project is collective. The experience gained from such collective learning can benefit future projects. This is represented by the hypothesis:

*Experience from executing a project can be applied for the effective implementation of future projects.*

Experience includes the existing knowledge of the individuals, the new knowledge created during team interactions, the insights gained and lessons learnt while executing the project. Such experience emerges from all the knowledge activities that occur while implementing a project, and is a result of the processes, rules, routines, policies and infrastructure of a software project organisation. Knowledge flows and exists within these processes and infrastructure, and needs to be leveraged to benefit future projects. This research therefore addresses the question:

*How can knowledge generated in the implementation of a software development project be leveraged and effectively reused in future software projects?*

While implementing a software project within an organisation, knowledge is generated as processes are executed. The organisation needs to identify the knowledge it requires to execute its software projects. Therefore, the research will explore the following questions:

*How is knowledge created and captured within software projects?*

*How is this knowledge stored, retrieved, transferred and applied or reused?*

Software projects require creative actions, heuristics and practitioner's experience as they encounter novel, unique and unstructured situations (see Sections 3.1 and 4.3). To address these situations, software project organisations need to identify and harness the tacit and explicit knowledge available to the project within its knowledge infrastructure. Therefore the research will also explore:

*What infrastructure is required to facilitate knowledge use within software projects?*

In addition, the research will also address how and where within the software projects the knowledge is used, and how this knowledge enhances the effective implementation of the project.

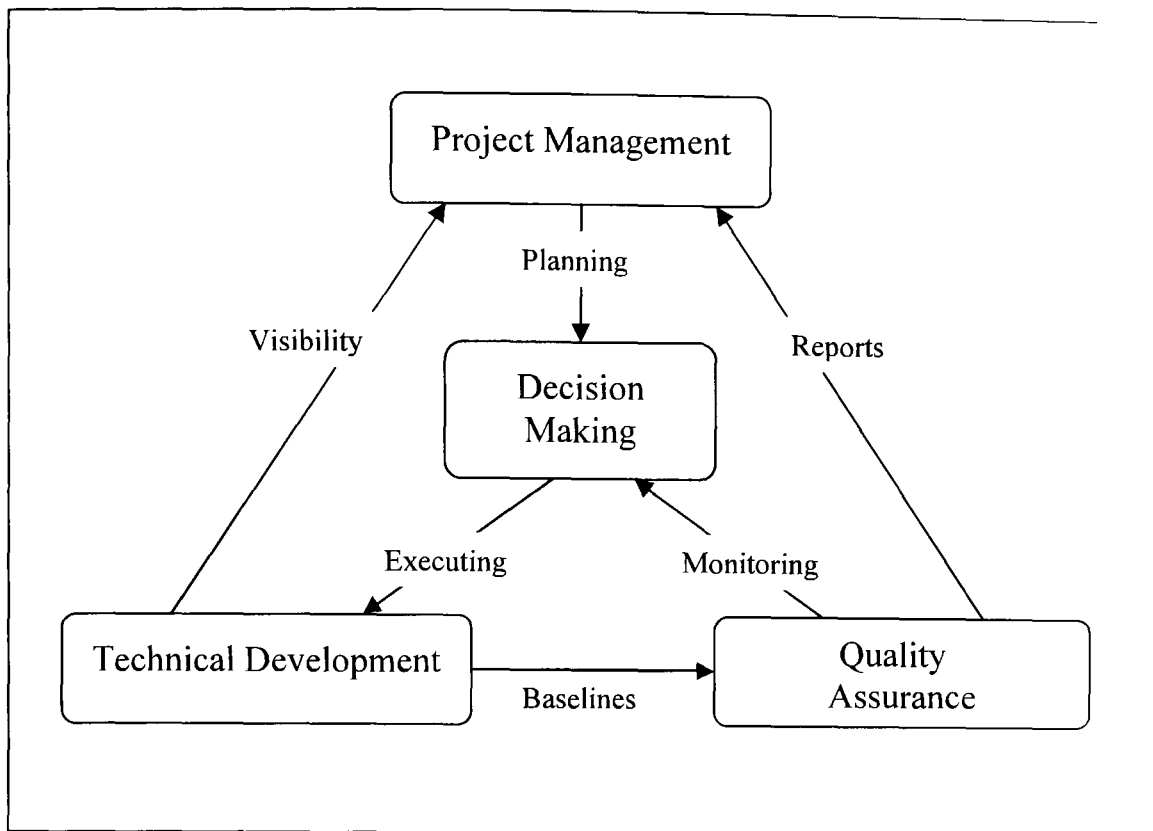
## **5.2 The Dynamic Feedback Model**

Several models exist, such as the Waterfall Model (Royce 1970), the Spiral model (Boehm 1988), prototyping and Rapid Application Development, which help conceptualise and understand the software development process. A model is a simplified description of a process and therefore helps in understanding the complexity involved within the software development process. As such, a model helps integrate the multiple functions of software

development such as planning, controlling, staffing, designing, coding, reviewing, testing and maintenance. Different models provide an emphasis on different aspects of software development. While the Waterfall and Spiral models depict software development as a linear activity with an emphasis on control, Prototyping and Rapid Application Development present software development in a more incremental manner, and agile methods highlight the evolutionary nature of delivery.

The previous chapter discussed how software organisations need to adopt a long-term approach to facilitate continuous learning and on-going interaction. A model that allows a continuous view and adopts a long term perspective of software development is the Dynamic Feedback Model (DFM), Dalcher (2002a, 2002c, 2003b, 2003c, 2003e, 2005a, 2005b). Dynamic models represent processes in contrast to static models that represent states, and this model therefore depicts the software process as a continuous description of relationships, interactions and feedbacks. Abdel-Hamid and Madnick (1991) state that ‘feedback is a process in which an action taken by a person or thing eventually affects that person or thing.’ Feedback is represented by a loop which is a closed sequence of causes and effects, or a closed path of action and information. An interconnected set of feedback loops depicts a feedback system, Richardson and Pugh (1981). The DFM underlines the relationships and interactions between different functional entities by depicting the feedback loops operating between them. The model, as depicted in Figure 4, focuses on four functional areas that are interlinked throughout the act of developing software. The DFM models the non-linear relationships amongst the functional areas of technical development, project management, quality assurance and decision-making thereby providing a continuous view of the development process.





After (Dalcher 2002a)

**Figure 4 Dynamic Feedback Model**

The technical area deals with the creation and continuous improvement of software development. The area recognises the changing needs and perceptions of the development process. The activities in this area include development and maintenance of the software, while also maintaining its functionality and utility. Experimentation, learning and discovery take place as the software goes from inception to evolution. The management area involves the planning, control and management of the software development process. The area also pertains to the strategy and operation of the project, and the key concerns revolve around identifying performance gaps, assessing progress, and allocating resources to accomplish tasks. As technical development and knowledge creation are on-going activities, the management area takes on a continuous view of the development effort. The area does not limit its focus to delivering a product, but to the continuous need for generating and maintaining an on-going flow of knowledge required for continuous development. The quality area is perceived as a dynamic dimension, which continuously responds to perceived mismatches and opportunities reflected in the environment. The area is concerned that the software product performs as expected by assuring the quality of the product developed and the process used to develop it. The decision-making area attempts to balance knowledge, uncertainty and ambiguity with a view to maximising the returns on an ongoing basis. The area provides a domain to conduct negotiations and trade-offs, and to accumulate new knowledge.

The four functional areas of the DFM are linked with a set of interactions and feedback loops that control the development of software. The basic loop in the dynamic system is the planning-control-visibility loop. This loop helps to plan and control the production, evolution and growth of the software in association with the project management and decision-making functional areas. The configuration control loop links control, baseline and monitoring that provide the link between monitoring and controlling. Monitoring provides feedback on the strategies implemented, and is the link between decision-making and quality. Monitoring makes the feedback system a closed-loop system, and ensures effective control while relying upon quality assurance techniques and feedback mechanisms to evaluate progress and quality. The reporting-planning loop links project planning, decision-making and quality assurance, and reports to management the trade-offs involved and quality of the software being developed.

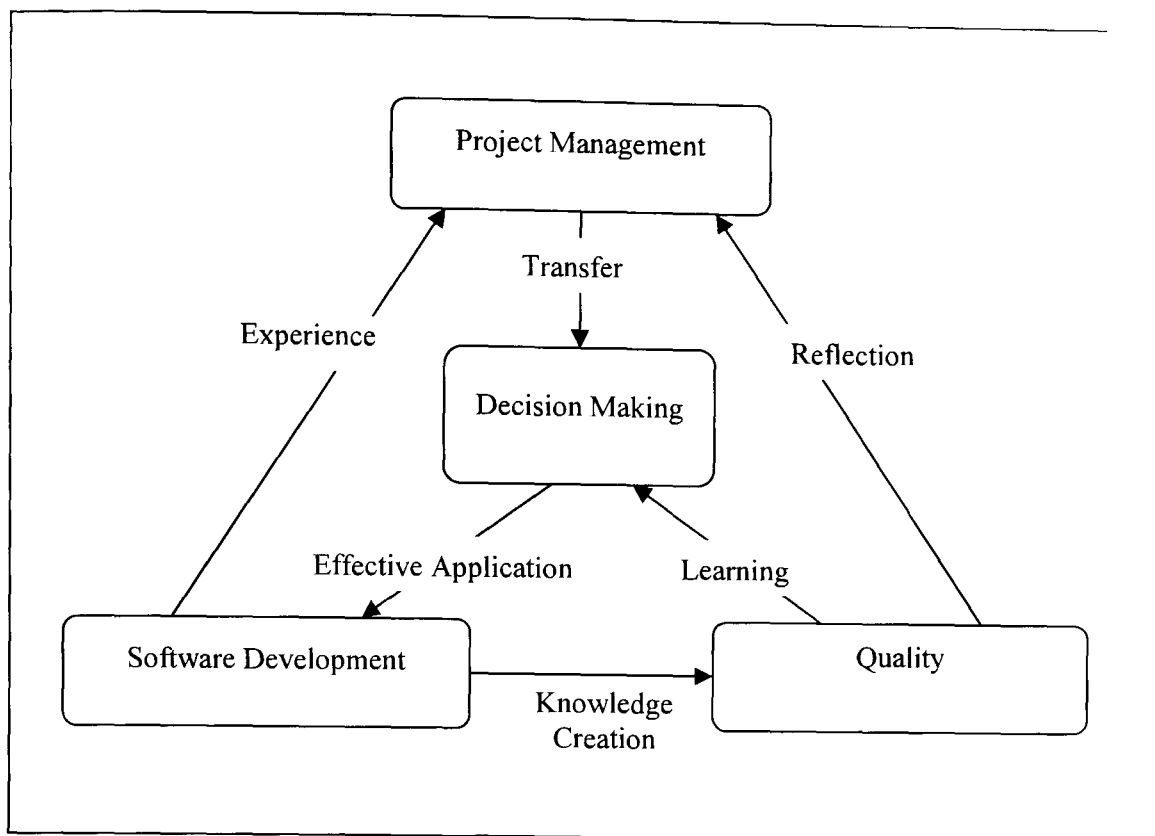
The feedback loops within the DFM help make sense of the dynamic nature of the complex interactions between the functional areas of software development. The DFM recognises and organises the four domains and helps implement their relationship and its long-term effects. Also, the DFM provides a project management perspective of software development. Particularly, the DFM provides the framework for the evolutionary delivery method of projects. The frequent interactions between the functional areas help facilitate the principles of evolutionary projects, which are to deliver something to the end user early, measure the added-value to the user in all critical dimensions, and adjust both design and objectives based on observed realities, Gilb (1988).

### **5.3 Adapting the DFM**

The feedback loops of the DFM depict the dynamic processes and interactions that occur while developing software. However, the existing version of the model does not specifically address the knowledge requirements and flows of the software development process. The DFM can be extended and adapted to incorporate the knowledge processes that underpin and support the project management and software development processes that the model depicts. Styhre's (2003) assertion that knowledge is 'processual' in nature and flows constantly, and is not located at any particular point in an organisation, applies to software projects. In fact, software projects in particular rely upon the creativity and tacit knowledge of the individual team members, Rubinstein and Pfeiffer (1980), and Lindvall and Rus (2003). Simon (1977) distinguishes between structured and non-structured situations, where repetitive and routine structured situations are addressed by

standard operating procedures, while unstructured situations require human judgement, insight and intuition. Software projects are by definition novel, unique and different. (see Section 3.1), and would benefit from the creativity, insight and knowledge, both tacit and explicit, of individuals. The DFM therefore needs to be adapted to provide the framework to facilitate the dynamic flow, transfer and application of such knowledge during the software development process. Thus the DFM is adapted to incorporate knowledge processes and flows as defined in this research, and is presented in Figure 5 below.

Also, Grady (1997) presents the plan/do/check/act (PDCA) process improvement cycle to improve the quality of software developed. Grady (1997) builds upon Shewhart's (1939) and Deming's (1986) work and applies the PDCA cycle to propose continuous improvement within the software development process, and consequently improve quality of the software produced. The fundamental principle of this cycle is iteration of the four steps or phases of planning, doing or implementing, checking, and acting or modifying the process to improve it before its next implementation. The PDCA cycle thus proposes that the software process improves continuously with each iteration thereby making the process more efficient to produce better quality software. This research evaluates the adapted version of the DFM model with Grady's (1997) plan/do/check/act (PDCA) software process improvement cycle to assess how the functional areas relate to the phases of the PDCA cycle. The research analyses and maps how the four functional areas of the adapted DFM relate to the four steps of plan-do-check and act. Since the PDCA cycle's ability to improve the software process and make it more efficient is assumed to be true, then mapping the functional areas with the PDCA steps should help make sense and establish that the knowledge flows as depicted by the adapted DFM will improve the effectiveness and quality of the software development process. The next four sections analyse the mapping of each of the four functional areas against the steps in the PDCA cycle.



**Figure 5 Hypothesis: Adapted DFM reflecting Knowledge**

### 5.3.1 Decision-Making Area and PDCA Plan Step

The individual team member's prior knowledge, both tacit and explicit, is brought into the decision-making domain, which lies at the heart of the process, and where commitment to the project is made and plans solidified. Prior experience helps make planning more effective, while knowledge support helps challenge assumptions made while planning and enable better decision-making. Decision-making enables making the appropriate choices during planning, thus providing similarities within this functional area of the DFM and the planning step of the PDCA cycle. The activities within this functional domain include an analysis of the impact the project will have on the business and technical environment along with the possible risks involved in implementing the project. The analysis views the goals, scope and functionality of the system being developed and how they fit or respond to the existing processes with which they are required to interact. Risk analysis is also conducted during this phase and it consists of the two traditional components of risk identification and prioritisation or projection. Identification tries to envision all situations that might have a negative impact on the project, while prioritisation involves analysing the possible effects and consequences of the risk in case it actually occurs. The project requires crucial decisions to be made in the design stage. High level design is the phase of the life cycle that provides a logical view of the development of the user requirements. Design

involves a high level of abstraction of the solution, through which requirements are translated into a 'blueprint' for constructing the software, and provides the architecture of the application and its database design. Decision making at this stage of the process helps transform the requirements into a set of software functions and a physical database structure. Scenarios are developed to test the acceptability of the design with relation to the requirements. The interactions amongst team members while performing tasks in this functional area create additional knowledge as per Nonaka and Takeuchi's (1995) model that can further be effectively applied, as suggested by Alavi and Leidner (2001), within the development area during the technical production of the software.

### **5.3.2 Technical Development and PDCA Do Step**

While the decision-making phase relates to the 'plan' phase of the plan/do/check/act (PDCA) phase of the software process improvement cycle, Grady (1997), the technical development area of the adapted DFM relates to the 'do' phase of the cycle. The technical area is where the technical activities of design, code generation and testing are performed to build and verify the software. The area deals with the detailed design phase where the high level design is broken down into modules and programs. A unit test plan is created for the conditions for which each program needs to be tested. The required programs are coded or translated into the programming language, and the programs are tested using the unit test plans. The technical area ensures that the integration plan is implemented according to the environments identified for integration. The area also ensures the maintenance and functionality and utility of the software apart from its creation and evolution. The decisions made in this area relate to the technical activities and provide confirmation of the design and suitability of the requirements. The knowledge created within this area flows into the quality domain to check mismatches in the functioning of the new software. The insights gained while developing the new product adds to the overall experience of both the individuals and the project teams, and can be fed back into the project management domain to be used in subsequent projects.

### **5.3.3 Quality Assurance and PDCA Check Step**

“Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that product quality is

meeting its goals” – Pressman (1997). The quality assurance area involves system testing which validates that the software developed meets the requirement specification. This phase relates to the check phase of the PDCA cycle, and identifies the defects that are exposed by testing the entire system. A series of tests are performed together, each with a different purpose, to verify that the system has been properly integrated and performs its functionality and satisfies the requirements. Therefore the quality assurance area provides verification of the decisions made and tasks performed in the technical area and confirm the decisions made during the design phase, and validating the requirements. The quality domain is an area of assessment that addresses mismatches and therefore provides opportunities for learning as mistakes are corrected. The learning that emerges from this area flows back to be incorporated into more decisions required to be made while the process of software development continues. Also, the learning and knowledge creation within the quality domain are reflected upon and fed into the project management domain to bring about more effective implementation of future projects.

#### **5.3.4 Project Management and PDCA Act Step**

Project management is about translating plans into actions. Experience and reflection help in the activities of the project management domain to ensure effective implementation of projects. The DFM helps feed such experience and reflection in the domain, which relates to the act phase of the PDCA cycle. The project management area involves project execution, and is also is where the user requirements are elicited and the problem defined. Proper requirements analysis and specification are critical for the success of the project, as most defects found during testing originate in requirements. In order to understand the customer requirements, the developers require insight into the domain of the business system and the technical concepts of the system to be developed. Knowledge is created while understanding the requirements by the interaction of the different team members, and also between the users and the developers. The project management area is where negotiation takes place between the users and developers of the software. Johnson (1993) defines negotiation as a process in which individuals or groups seek to reach goals by making agreements with each other. Software development requires that users are involved in the development of the software. A clear understanding is needed between the users and developers to build the software, and this understanding is established through dialogue and negotiation. The formalisation of such an understanding usually results in the form of proposals and contracts. Knowledge is created in the interaction between the developers and users while trying to understand the user domain and defining the software

specifications. This created knowledge enhances the already existing knowledge and provides the perspective for decisions made to implement the current and subsequent projects, and is transferred to the decision-making domain.

### **5.3.5 Summary – Adapting the DFM**

The adapted version of the DFM provides a perspective of the knowledge management process support required by the project management and software development processes. The model incorporates the knowledge management processes of knowledge creation, transfer and application. Knowledge storage is a static aspect of the knowledge management processes. Also, the process of storing knowledge converts tacit knowledge into explicit. The adapted model recognises software developments requirement and reliance on tacit knowledge, and provides a framework which underpins the tacit knowledge support of the developmental process. However, it is important to note, that while adapting the DFM, an attempt has been made to identify how the knowledge management processes relate to the software development processes and activities within and between the functional areas of the previously published DFM. In doing so, the research has identified the areas where there is greater emphasis of a knowledge management process activity rather than an exclusive flow of that particular activity.

The dynamic nature of the model addresses and facilitates the dynamic nature of knowledge. Reinforcing Styhre's (2003) concept of constantly flowing knowledge the feedback loops ensure the flow of knowledge and globalise it within the project management and software development processes. The flowing knowledge can be utilised at anytime, at any point, and within any functional area of software projects. Therefore, this version of the DFM meets knowledge management's objective of getting the right knowledge to the right person at the right time.

## **5.4 Conclusion**

This chapter develops the hypothesis and research question, and the related questions that stem from the main question, addressed by this research. The chapter identifies, extends and adapts a published framework that helps support, and attempts to resolve some of the issues that are at the core of this research. While previous frameworks present in the literature support knowledge management within permanent organisational structures, this

framework can be extended to facilitate the dynamic nature of knowledge and its management within project environments. The framework is matched with the PDCA cycle of the software processes improvement cycle to evaluate and assess the capabilities of the model presented within the framework.

The Dynamic Feedback Model depicts the continuous nature of software development in contrast to the typically static representation of the process. This research also evaluates and assesses how the knowledge management processes can be facilitated by the model to support software projects. The framework, in particular, addresses an important requirement of software development projects by enabling the support of tacit knowledge transfer and application. Supporting the use of tacit knowledge distinguishes this framework from others which primarily address and support the use of explicit knowledge only. The adapted framework thus becomes the embodiment of the research hypothesis



## Chapter 6 Research Framework

The previous chapter discussed the research problem and presented a model that embodied the characteristics and essence of the research question. This chapter discusses an overview of research methods and analyses the research design required to meet the aim and objectives of the research.

### 6.1 An Overview of Research Methods

*There is no method peculiar to philosophy – Karl Popper.*

Popper (1959) propounded the above thesis with regard to practitioners of a method peculiar to philosophy, but the thesis applies to other disciplines too. Simply stated, the research problem determines the method required. Critically examining the various proposed solutions to a clearly defined problem results in an appropriate method to explore the problem. However, such an approach is often lost within the perception of a dichotomy between quantitative and qualitative research methods.

Quantitative research is typically viewed as being a systematic approach to investigating a research problem. An example considered to depict a quantitative research method involves testing a hypothesis formulated on the basis of a theory. Testing the hypothesis requires data that is collected systematically with a 'scientific' approach, Bryman (1989). Qualitative research, on the other hand, is considered to be more 'interpretive', where the emphasis of the research is on the people being studied within a context. Quantitative and qualitative research methods are considered to be distinct in their approaches to data collection, with quantitative research considered to collect data that is more generalisable, while qualitative research lays an emphasis on the researcher collecting data at the source of what is relevant to the research problem.

However, quantitative and qualitative methods are more than a set of data collection techniques. Rather, they are approaches to addressing problems and seeking answers to those problems within a certain mindset. While a quantitative approach would attempt to systematically measure the degree to which a feature or characteristic maybe present, a qualitative approach would attempt to identify, study and interpret a phenomenon within a context. Qualitative research fundamentally depends on observing and interacting with

individuals in their environment, and is considered to be naturalistic, ethnographic and participatory, Kirk and Miller (1986). The emphasis in qualitative research is on understanding the participant individual's actions and behaviour within their environment and on their own terms. Therefore, in this type of research the collection and analysis of data needs to be sensitive to the nuances of what the individuals say and to the context in which their actions take place. In contrast, quantitative research lays an emphasis on the systematic collection of data to test a hypothesis based on a theory. In doing so, quantitative research relies upon concepts that can be measured, operationalised, generalised and replicated.

Hypotheses contain concepts that quantitative research requires to be measured in order for them to be systematically tested. Bryman (1989) states that the process of translating concepts into measures is termed operationalisation, and that these measures are treated as variables that are the attributes on which variability is exhibited. Bryman (1989) also states that many hypotheses contain implicit or explicit statements about cause and effects, that is, in showing how concepts come to be the way they are, or causality. The emphasis on causal effects relies upon the relationship between independent and dependent variables in quantitative research design. As quantitative research is considered to generate data that is systematically collected, it is expected that the results and inferences drawn from the data can be generalised beyond a specific or sample investigation to a broader population, and replicated. Replication checks researcher bias and enhances confidence in the findings that are verified.

Qualitative research begins with a set of loose concepts and ideas that progress towards the theoretical aspects of the research based upon the perspectives and inferences drawn from the concepts examined. In doing so, qualitative research involves the phases of invention, discovery, interpretation, and explanation. Invention denotes the phase where the research design is prepared, while discovery relates to the data collection phase. Interpretation is the phase of evaluation and analysis, while the explanation phase presents the new understanding that emerges towards the end of the research. Therefore, even though qualitative research is developed within theoretical frameworks, it allows for a more flexible research design where the procedures are not necessarily standardised, but as rigorous and demanding as those required by quantitative research.

Also in contrast to quantitative research that attempts to reduce individuals, groups and their environment into attributes and variables, qualitative research views them as a whole

and attempts to study them in the context of their past and existing situations. Qualitative research provides rich descriptions and explanations of processes and interactions in identifiable local contexts that enable the researcher to observe chronological flow, and see precisely which events led to which consequences, and thus derive explanations. Qualitative research is inductive and aims to develop concepts, insights and understandings from patterns in the data, rather than collect the data to assess the prior concerns developed in the hypotheses, models and theories. Glaser and Strauss (1967) refer to grounded theory as the inductive theorising process as it is derived from and based on the data. Lofland (1995) describes such theorising as emergent analysis and states that the process is creative and intuitive as opposed to mechanical. Qualitative research is concerned with the meanings and perspectives individuals attach to their environment and situations. Qualitative research methods enable the researcher to stay close to the phenomenon being examined, and are designed to study and interpret the data collected and action of the individuals being observed. Qualitative researchers obtain first-hand data by observing individuals performing their daily tasks, listening to what they say, and examining the documents they produce. Qualitative researchers are then able to identify and reconstruct the perspectives and patterns of action and interaction observed, to gain an understanding of the phenomenon and concepts being researched. Miles and Huberman (1984) state that “findings from qualitative studies have a quality of ‘undeniability’, as words, especially, organised into incidents or stories have a concrete, vivid, meaningful flavour that often proves far more convincing to a reader than pages of summarised numbers.”

The above discussion suggests that quantitative and qualitative research methods are distinct approaches to addressing research problems, issues and concepts. To sum up, quantitative research aims to collect data systematically to address prior concerns, and generalise the results to a larger population. Qualitative research is more flexible and is concerned with the perspectives of individuals being studied within a context, and it leads to a new understanding that emerges towards the latter stages of the research. Both approaches reinforce the term *methodology* that refers to the way in which researchers address problems and seek answers. The approaches are not mutually exclusive and research design often requires that a combination of qualitative and quantitative methods be applied. Qualitative researchers often include quantitative procedures in their investigations, while quantitative researchers sometimes collect additional qualitative material to supplement their research. Goldstein and Goldstein (1978) emphasise this by suggesting that intuition, creativity and subjectivity are an integral part of ‘scientific’ discovery.

## **6.2 Research Designs**

Research design is the overall structure of the research approach and provides the framework within which data is collected and analysed. The research design addresses the logic of the study rather than its logistics, and prepares the plan for collecting and analysing data to provide answers to the research questions. The following section discusses the more commonly used research designs.

### **6.2.1 Experimental Research**

Goldstein and Goldstein (1978, p206) state that experimental research is “accepted as one of the primary criteria of scientific truth.” Subjecting explanations to experimental test is considered a distinguishing feature of the scientific approach. Experimental research is conducted within a conceptual framework and is an attempt by the researcher to maintain control over all factors that may affect the result of the experiment. Experiments help in establishing claims about causality as the hypotheses is tested by reaching valid conclusions about relationships between independent and dependent variables. Within experimental research design, the researcher attempts to determine or predict what may occur in the experimental setting after neutralising the effects of certain factors.

An important aspect of experimental research design is its validity, which could be internal and external. Internal validity is important if the experiment treatment makes the difference rather than external factors, while external validity is concerned with what populations, settings, treatment variables, and measurement variables can the observed effect be generalised. Goldstein and Goldstein (1978) are of the view that the outcome of an experiment must make a difference in order for it to be considered a test of theory. The outcome could lead to a greater degree of either confidence or doubt regarding the assumptions and beliefs being tested.

### **6.2.2 Survey Research**

Bryman (1989) states that ‘survey research entails the collection of data on a number of units and usually at a single juncture in time, with a view to collecting systematically a body of quantifiable data in respect of a number of variables which are then examined to

discern patterns of association.’ Surveys are a method of gathering information from a ‘sample’ of individuals, where ‘sample’ is a fraction or portion of the population being studied. The sample is representative of the population to be studied and its size is determined by the purpose of the study. The surveys outcome depends upon the choice of the sample and, therefore, the methods of choosing a sample assume significance in the survey research design. Samples are chosen by the random or probability sampling method which attempts to give each unit of the population a chance of being included in the sample. Probability sampling attempts to eliminate selection bias whereby certain units could be over-represented, and also seeks to reduce sampling error, which is the difference between the sample and the population.

Surveys are conducted by questionnaires and interviews. Surveys conducted by questionnaires often suffer from the problem of non-response bias, wherein a significant part of the sample population does not provide a response. Also, as data is usually collected at a single juncture during the research phase, surveys could pose problems in determining cause and effect while inferring causal relationships from their findings. However, despite these difficulties, surveys provide a snap-shot view of the phenomenon being studied.

### **6.2.3 Case Study**

A case study is a detailed examination or investigation of one or a small number of ‘cases’, which could be individuals, groups, organisations or other social units. A case study attempts to study a phenomenon within its context, usually under natural conditions, enabling the phenomenon to be understood in its environment. Yin (2003b) states that case study is the design required when the phenomenon under study is not readily distinguishable from its context. Also, a case study differs from other research designs because its focus of attention is the individual case, or cases, and not the whole population of cases. As such, the focus and priority of a case study may not be on generalising its findings but on understanding the complexity of the particular cases. A case study provides the possibility of combining qualitative and quantitative research design, though in most cases it is identified as a type of qualitative research approach. The richness of context in a case study means that the research design will likely have more variables than data points, and that the study cannot rely on a single data collection method but will likely need to use multiple sources of evidence. Combining the advantages of qualitative and quantitative

approaches enables a case study to gain deep theoretical insights and understandings of the issues surrounding the phenomenon being studied.

Further, Yin (2003b) states that case study research is appropriate when researchers are required to 'define research topics broadly and narrowly, to cover contextual or complex multivariate conditions and not just isolated variables, and to rely on multiple and not singular sources of evidence.' Yin (2003a) sums up the above essence by stating that a case study is a comprehensive, empirical research strategy that 'comprises an all-encompassing method that covers the logic of design, data collection techniques, and scientific approaches to data analysis.' However, despite being an all encompassing method as stated by Yin (2003a), case studies can be limiting in generalising their findings. Replication is possible in multiple case studies, but Bryman (1989) states that when a large number of cases are involved, some of the distinctiveness of case study research is lost.

A case study can be broadly categorised as exploratory, descriptive or explanatory. An exploratory case study is aimed at defining the questions and hypotheses of a subsequent study or at determining the feasibility of the desired research procedures. A descriptive case study presents a complete description of a phenomenon within its context. An explanatory case study presents data bearing on cause-effect relationships, that is, explaining how events happened.

#### **6.2.4 Action Research**

Action research design provides a framework within which a researcher becomes a part of the field investigation. The researcher makes suggestions and observes the impact of the implementation of those suggestions on the problem being studied. Action research is often a 'participative' approach in which the action researcher collaborates with practitioners to conduct the research investigation to the problem domain to gain a new understanding. The relationship between the researcher and the practitioner subjects is an important aspect of action research as in the case of an organisation, several members of the organisation at various levels may participate in the research process. The nature of the relationship leads to a distinctive research design that involves the applications of the research findings as the research provides focused efforts to improve the quality of an organisation and its performance.

Action research is iterative or cyclical in nature starting with conceptualising a problem, operationalising its solution, and moving through several interventions and evaluations. Initially, an understanding of the problem is developed and plans are made for an interventionary strategy. The intervention is implemented and during the intervention observations are collected. Revised interventions are carried out, and the cyclic process repeats, continuing until an implementable solution or understanding of the problem is achieved. Such iteration makes reflection an integral part of the action research design.

### **6.3 Methods of Data Collection**

Methods refer to the techniques or manner in which data is collected. Perry (2004) states that the three principles of data collection for a good research design are using multiple sources of evidence, creating a database, and maintaining a chain of evidence. Using multiple sources of evidence helps in reinforcing and validating the findings through the process of triangulation, which Yin (2003a) refers to as 'converging lines of inquiry.' The data collected while implementing the research design needs to be stored in a database in the form of notes, documents, questionnaires, tabular materials or narratives. Such storage helps maintain and develop a chain of evidence between the issues at the beginning of the research, the data collected, and the conclusions drawn. The following section discusses the commonly used methods of data collection, which Perry refers to as 'sources of evidence.'

#### **6.3.1 Observation**

In the method of data collection called observation, the researcher observes individuals or participants in natural or structured environments. Naturalistic observation is done in real-world settings while laboratory observation is done in a structured environment set up by the researcher. Observation can be broadly categorised as structured, quantitative, qualitative, direct and participant, and in all categories it enables the researcher to view behaviours and environmental conditions relevant to the research. Observation relies upon the recording of notes of what is being observed. As such, if conducted by a single researcher, the research findings can suffer from bias and a singular perspective. However, the advantage of observation is that often the participant individuals are unaware that they are being observed, and therefore the data collected is naturalistic with the behaviours

concerned. Thus, observation is a reliable and valid method when the research design requires a focus on *actual* behaviour and gathering specific facts.

Participant observation provides an opportunity for the researcher to gain access to events or groups, and perceive 'reality' from the viewpoint of an actual setting within the context of the research. Yin (2003a) states that participant observation also provides the researcher the opportunity of 'active' research where he or she is able to manipulate minor events.

### **6.3.2 Interviews**

Interviewing is a data collection method in which the researcher asks an individual questions. However, in some exceptional and rare situations, the questions maybe addressed to a group. Through the questions, the researcher is able to follow a line of inquiry required by the research design, while attempting to ask the questions in an unbiased manner. Interviewing requires subtle skill as the researcher attempts to probe for information relevant to the research, while also trying to establish trust and rapport with the interviewee. Interview as a singular research method can suffer from the problem of bias, poor recall and inaccurate articulation, and therefore, a good research design needs to corroborate interview data with data collected from other sources.

Interviews are generally viewed as open-ended or focused interviews. Open ended interviews involve descriptive questions that enable the respondents to talk about issues important to them and give their opinion and perspective. Such interviews tend to result in eliciting rich insights into occurrences and issues, as well as leads to other sources of evidence, relevant to the context of the research. Focused interviews are conducted by asking standardised questions, possibly based upon a previously agreed interview protocol. Questions in a focused interview are structured and maybe open-ended or close-ended. Close-ended, structured interview questions tend to look similar to a questionnaire that is administered verbally.

### **6.3.3 Questionnaire**

A questionnaire is a data collection instrument that is filled out by research participants called respondents. A questionnaire requires participants to respond to questions and/or statements related to the research objectives. Questionnaires are generally designed in a



clear, precise and relatively short and simplistic manner to make them understandable to the participants after considering their demographic and cultural characteristics. Questionnaires can also be designed to contain open-ended or closed-ended questions. Open-ended questions are commonly used in exploratory research designs and enable participants to answer qualitatively in their own words. Close-ended questions on the other hand, are more common in confirmatory research where the participant's answers provide quantitative data based on the researcher's response categories. Response categories require the participants to decide upon rankings, rating scales or checklists while responding to the questions. Often, the design of a close-ended questionnaire requires the wording of some items to be reversed to prevent a response set, which is the tendency of a participant to respond in a specific direction to items regardless of the item content.

Questionnaires are often used in research designs that require studying a large sample population across a distributed geographical area. Uniform questions are considered to reduce bias and are easy to analyse. However, questionnaires suffer from low response rates and an inability to probe responses. Questionnaires are structured instruments and therefore do not allow the participants much flexibility while responding to the queries. Since questionnaires avoid the researcher's direct contact with the participants, they eliminate the 'flavour' of the response required for sensitive research issues. Therefore questionnaires are more appropriate for collecting quantitative, factual data rather than qualitative data that requires viewing phenomenon within a context.

#### **6.3.4 Documentation and Archival Records**

Documentary evidence provides a good source of data collection for research purposes. Such evidence is often present within letters, reports, minutes of meetings, agendas, announcements, summaries, or media related records. Apart from the regular organisational sources, documentary evidence is also available in personal diaries, calendars and contact lists, and can be broadly categorised as recent or historical.

Documentary evidence is considered non-reactive and therefore presumably without the bias which is present in evidence gathered from respondents in interviews and questionnaires, or through observation. Documentary evidence covers a longer time span in the organisation's existence and is therefore a good provider of information regarding organisational change. Also, documentary information provides access to information

about individuals who are not accessible through conventional approaches. However, being non-reactive, documentary evidence is liable to be interpreted diversely and is therefore valuable when used in conjunction with other data collection methods such as interviewing and observation. Yin (2003a) states that for case studies, the 'most important use of documents is to corroborate and augment evidence from other sources.'

## **6.4 Research in Information Systems and Computing**

“Information systems as a discipline is concerned with the development and use of information systems by individuals, groups, organisations and society, where usually those information systems involve the use of computers,” Oates (2006). Glass et al (2004), break down the computing field into three common academic sub-divisions of computer science, software engineering and information systems. Glass et al (2004) state that computer science examines topics related to concepts at a technical level of analysis formulating processes, methods and algorithms using mathematically-based conceptual analysis, while software engineering address the same problems using non-mathematically-based conceptual analysis. According to them, both computing science and software engineering do not rely upon reference disciplines. However, information systems research on the other hand relies upon a variety of reference disciplines, and examines topics related to organisational concepts and behavioural aspects of information technology development, deployment, adoption and usage. As such, information systems research is usually conducted in organisational settings within a behavioural context, and relies upon research and theories of other related disciplines. Information systems research addresses how information technologies are developed, implemented and used in the ‘real world’ settings, and as such look towards an empirical assessment of systems. Dalcher (2003b) states that since empiricism relies on direct observation or immediate experience gained through the traditional five senses, empiricists employ the practitioner’s viewpoint, and “empiricist theories based upon experience are more open to correction, development and the test of further experience.”

Knowledge management initiatives require cultural and infrastructure issues to be addressed within organisational settings. This research attempts to provide a conceptual understanding of how knowledge management processes support project management and software development processes. To achieve this, the research draws upon theories and references from computing, management, software engineering, information systems and

organisational behaviour disciplines among others, and therefore it requires an empirical inquiry to investigate the issues involved.

## **6.5 Research Design**

Section 5.1 established that this research needed to examine how knowledge is created, stored, retrieved, transferred and applied or reused while implementing software projects. This required the research to be able to have access and observe the implementation of software projects. Section 5.1 also discussed the need for the research to explore how and where within software projects is the knowledge used, and what infrastructure and capabilities is required to facilitate and support such knowledge use. This required the research to be able to observe the functioning and interactions of project teams, and also be able to interact with individuals who were implementing and executing software projects.

Qualitative research methods, as discussed in Section 6.4, provide many avenues to study the above mentioned interactions within an organisation, including interviewing organisational members and observing their behaviour. The case study approach, in specific, is often used and suggested in conditions where several elements and multiple dimensions of a subject need to be studied exhaustively (Alavi and Carlson 1992, Benbasat et al 1987, Eisenhardt 1989, Yin 2003a). Yin (2003a) describes a case study as “an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.” This definition captures the characteristics of the study examining the role of knowledge support for software development projects as undertaken by this research. This was particularly significant as knowledge management processes are context related, Nonaka and Takeuchi (1995), and therefore needed to be observed, studied and examined within their real-life context.

### **6.5.1 The Case Study Organisation**

The research conducted a case study at an organisation that develops and maintains software applications using CMMI Level 5 and ISO 9001 quality processes. The organisation has developed software applications and solutions for more than thirty-five years and has mature software processes as implied by the quality certifications. The research benefited by being able to explore how these mature software processes are able

to leverage the knowledge that is created. The organisation implements software projects on a regular basis and therefore provided an ideal real-life setting to explore the research questions. The organisation operates across more than 40 countries and has more than 150 offices worldwide. The organisation implements software projects with an annual sales turnover of over \$4 billion, and currently employs more than 108,000 individuals.

The case study was conducted within the organisation over a period of twenty-six months. The length of the case study ensured that the researcher was able to revisit and revalidate certain assumptions, and clarify issues, observed during project implementation. During preliminary discussions with the organisation it was agreed that the researcher would have complete access to software projects currently being implemented. This access would include being able to observe project team meetings, functioning and interactions, including interactions with clients. Team meetings observed included project start-up, design, weekly review, and closure. Project team functioning included software development at the organisation and on-site work conducted by team members at the client's site.

The access to software projects agreed with the organisation also included permission to conduct open-ended interviews with individuals with varying levels of seniority and experience. This resulted in the researcher being able to carry out approximately one hour long interviews with a cross-section of individuals ranging from software developers to an Executive Vice President. Thus the researcher was able to obtain insights into software project implementation from diverse perspectives including a developer's technical perspective, a project manager's project management perspective, a Software Engineering Process Group (SEPG) member's perspective of an advisory role that facilitates and shapes software project implementation policies, and senior management's strategic perspective on implementing projects. Thus the access agreed provided this research with a special and unique opportunity to gain insights into the work practices and development processes of one of the largest globally distributed organisations in the world.

### **6.5.2 Structuring the Research**

Goncalves (2005) states that "the main challenge in applying KM to project management is that KM is overwhelmingly a cultural undertaking." Before implementing a knowledge management initiative it is imperative to identify what knowledge is required by individuals and what mechanisms and practices are required to be implemented to get them

to share such knowledge. Khalifa and Liu (2003) identify leadership, culture and a KM strategy as the infrastructure required to successfully implement a knowledge management initiative. Appropriate norms and values motivate knowledge sharing and collaboration. This is particularly important for motivating the sharing of tacit knowledge, which according to O'Dell and Grayson (1998) is not likely to be transferred through predefined formal means. To foster a supportive culture for knowledge management, individuals must be able to appreciate and recognise the value of knowledge management initiatives, Alavi (1997).

This research examined the knowledge management infrastructure and processes and how they help facilitate the implementation of a knowledge management initiative. The knowledge sharing practices and how they are aligned with the knowledge strategy of the organisation were observed. The research identified the type of knowledge strategy adopted by the organisation. Hansen et al (1999) argue that organisations should focus on just one type of strategy. However, software organisations encounter situations where knowledge is very context dependent and therefore require a strategy that facilitates the sharing of explicit as well as tacit knowledge.

The aim of this research was to investigate how knowledge generated while implementing a software project can be leveraged and effectively reused in future projects. Some project knowledge is captured within the documents produced while implementing the project such as project plans, requirements, design and testing specifications, and solving problems encountered. Often, post-project reviews are conducted to identify the lessons learned and for post mortem analysis of the software product and processes. However, by the end of the project it is possible that some knowledge created during implementation may already be lost. The research analysed whether the mature standardised software processes of the organisation were sufficient to leverage all knowledge within the organisation, or were additional knowledge processes required to ensure that software projects were being implemented effectively. The research also analysed the knowledge processes being practiced within the software processes, and discussed this issue in detail during open-ended interviews.

The organisation has global clients and dispersed software development centres. Therefore the organisation presented an opportunity to study the knowledge activities related to collocated as well as distributed project teams, and how these teams interact with the end users of the software being developed. The research observed team interactions during

project implementation and identified instances of knowledge sharing behaviours. The research analysed the knowledge management infrastructure being used and knowledge processes being practiced

The research observed the work practices of project teams for tasks such as requirements analysis, design, implementation and quality assurance. Besides being able to attend meetings related to the mentioned activities, the researcher was able to spend time within the open plan offices of the organisation to observe the informal interaction and work practices. Thus, the researcher was able to observe and gain insights into the daily activities of individuals at the organisation.

As discussed in Section 2.4, information technology support plays an important role in facilitating knowledge management processes. The research observed the information technology support the organisation possesses for its knowledge management processes. Based upon their usage and characteristics, the information technologies can be grouped within the categories of knowledge repositories and libraries that store knowledge in the form of documents, information technologies that support the collaborative work practices of individuals, including distributed work practices, and cartographic information technologies that enable mapping and categorising knowledge such as core competencies and expertise within the organisation to individual expertise.

The objective of this research was to observe the functioning of project teams while implementing software projects. The team interactions were observed during the different stages of software development processes. Team interactions provided the researcher with insights into project managers' perspectives on leadership, mentoring and leading by example as role models. Observations of team interactions and functioning revealed instances of decision-making, collaboration, trust, initiative, problem-solving, and knowledge sharing practices amongst individuals. These observations are significant for the study as Lindvall and Rus' (2003) state that 'KM is unique because of its focus on the individual as a consumer of knowledge and as bearer and provider of important knowledge that could systematically be shared throughout the organisation.'

### **6.5.3 Fitting Knowledge Management within Project Management and Software Development Processes**

Typically the stages of the software development process are problem definition, requirements analysis, and design, the implementation of which includes coding and testing, and maintenance thus involving conception, delivery and operation of the system. These software development stages are coordinated within the project management processes of planning, execution and closure.

The research observed the knowledge processes of the team members during the planning stage. This stage typically involved defining the life-cycle process to be followed, estimating the effort and schedule, preparing a detailed schedule of tasks, planning the quality control, and also planning for possible risks. The team members bring their past experience to help in the effective and efficient execution of the tasks during this stage. The research observed how this experience is applied and new knowledge created during the interactions of this stage. As mentioned previously, this stage culminates with the preparation of a project plan. The research observed how knowledge created during this stage was captured within the project plan. Important lessons regarding scheduling and effort estimation would have been learnt from previous projects. The research observed how these lessons benefit the tasks at hand during this phase of the current project. Based upon the observations made during the phase, the researcher interviewed team members. In certain instances where the team size was large and access to all team members was not possible, the researcher interviewed key team members who played a leading role in team discussions and deliberations. Most interviews conducted were open-ended interviews to gain insights into the perceptions, thoughts and views of the individuals. However, again during certain instances of time constraints, focused interviews were conducted based on the observations made to gain insights into the key issues.

The project execution stage involved executing the project plan, tracking the status of the project, and making corrections whenever the project performance deviates from the path laid down in the project plan. Tracking and controlling the implementation of project processes present opportunities for learning and knowledge creation. As deviations are detected and corrected, their causes are analysed and learning takes place. The research observed how the organisation manages the new knowledge that emerges from learning. The deviations that occur during the monitoring and controlling phase provide a better understanding of the problems of implementing projects. During interviews the researcher

attempted to gain an insight into how the team members interpreted these deviations. Significant features of this phase of the project management process were frequent review meetings. Such meetings involved team interactions that have to decide upon corrective action. Therefore, these meetings provided a frequent opportunity for knowledge creation within this phase. Also, this was the phase where the software product was developed and tested. Developers often use their heuristic skills while programming, while testing helped uncover mistakes and controlled the quality of the software. Therefore, knowledge was created at different times and at multiple places during the project execution stage, which the researcher attempted to identify and analyse.

The project closure stage is the last stage of the project management process and involves the systematic wind-up of the project. This is the stage that aims to learn from the experience so that the development process maybe improved. The activities in this stage are team activities that involve analysing the post project data, reviews, and the lessons learnt from implementing the project. Quantitative analyses are performed on the project metrics and the new process assets discovered. Post project reviews are conducted and collective project learning emerges. The research examined the organisations post project review process and analysed possibilities of improvements to avoid loss of project implementation knowledge. These included the possibility of conducting more frequent reviews while implementing the project, and post project retrospectives.

Though learning and knowledge are ‘supposed’ to be a result of tasks executed during the last stage, they occur and emerge through all three stages of the project management process. Subsequent to the observations and interviews conducted during the software development and project management processes, the researcher reviewed the knowledge activities undertaken by the organisation while executing these processes. The researcher analysed how knowledge flows during the software and project management processes of the organisations. Also, the researcher analysed how the captured knowledge is stored and applied for further use. The analysis involved reviewing the knowledge management infrastructure and process capabilities support for the software and project management processes of the organisation.

#### **6.5.4 Data Collection and Analysis within the Research Design**

The researcher interviewed individuals who had worked on multiple projects to gain an insight into how their prior experience benefited them in their future projects, and



enhanced the effectiveness of the project teams. During the course of these interviews an attempt was also made to get the individuals to identify practices and policies of the organisation that helped them perform their tasks more effectively in later projects.

A significant feature of the observations conducted of team meetings and processes was that at the end of the session the researcher was asked to address the team members. The researcher would thereafter make a presentation about the research being conducted and the reason for his presence. This inevitably resulted in an interactive discussion lasting a few minutes. At the end of the discussion the researcher would present to the team members the recorded observations of the session in an attempt to ensure that the observations were accurate and depicted the nature of the meeting or session. The team members were again asked if they had any further questions or clarifications. In this manner the researcher was able to confirm the observations made were accurate to the extent possible.

As is evident from the above discussion, the data was collected while conducting the case study for a period of over two years through interviews and observations. The interviews were conducted on-site and were initially structured around the history and emergence of knowledge management as a concept and practice within the organisation, and thereafter along the theoretical propositions of the study to examine knowledge management practices and capabilities, and software and project management processes. Miles and Huberman (1994) describe the analytic phase as consisting of three concurrent flows of activities that are data reduction, data display and conclusion drawing. Data reduction refers to the process of selecting, simplifying, and abstracting the raw data that the researcher has in forms of filed notes and transcribed interviews. Data reduction is part of analysis because the researcher makes explicit choices of what categories to use, what sources to include, and what data to summarise. Data display is an organised spatial way of presenting the data systematically to the researcher in a form that helps the researcher see what is happening. Again, the process of data display is part of analysis, since deciding what data to present and in what form affects the result space. Conclusion drawing, is deciding what things mean, and is also a part of analysis. Conclusions first appear as vague impressions that subsequently are validated as the analysis proceeds. Qualitative data analysis is therefore an iterative exercise that also includes the data collection phase.

This research has made use of data reduction, data display and conclusion drawing. The data was read, categorised, conceptualised and interpreted in an iterative manner. The

research drew categories from the key words of the transcribed interviews and observations. Attributes were assigned to each category, and the categories were iteratively reviewed on the basis of their attributes. This resulted in categories determined by their attributes, which enabled the research to conceptualise and interpret the empirical evidence and evaluate the emerging themes. The research was thereafter able to draw conclusions based upon the theoretical foundations presented in the previous chapters. The detailed process of data collection and findings are presented and discussed in the next chapter while the conceptualisation of this research design is presented below in Figure 6.

## **6.6 Conclusion**

This chapter presented an overview of the research methods and the specific methodology adopted by this research. The chapter discussed the quantitative and qualitative research approaches, and the different aspects of research design and data collection methods. Research within the discipline of information systems and computing was also discussed, and the research design for a case study to address the research question was developed. The sections of this report that follow, present the findings, evaluation, analysis and conclusions based upon the research methods and design described in this chapter.

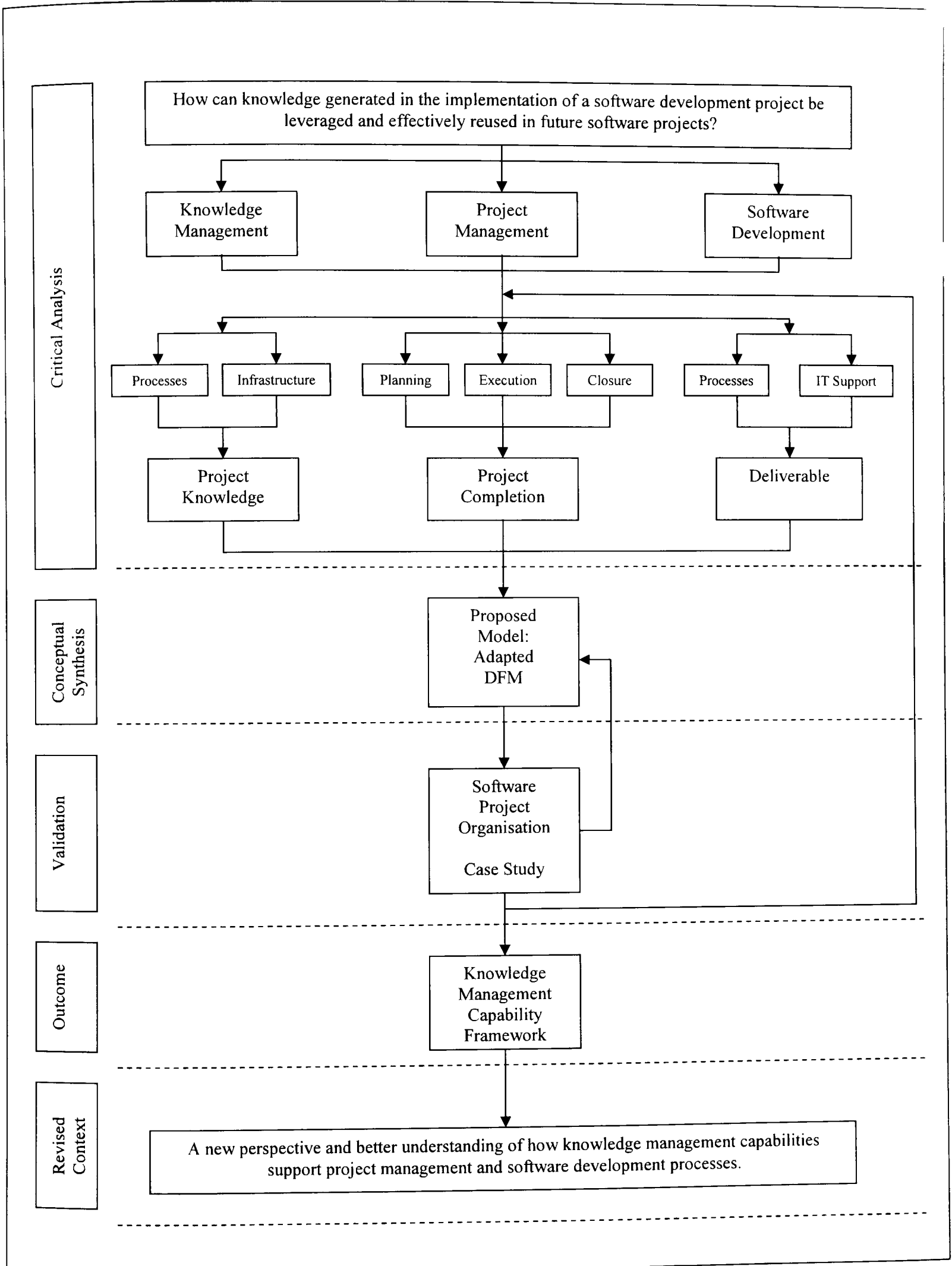


Figure 6 Research Design

## **Chapter 7 Data Collection and Findings**

The previous chapter discussed an overview of the research methods and the specific methodology adopted by the research. This chapter discusses how the data was collected and presents the findings derived by the research based upon implementing the research methods and design.

The research followed Yin's (2003) Case Study protocol while conducting the case study at a large CMMI Level 5 software project organisation. The researcher was able to interview a cross-section of individuals at the organisation that included an Executive Vice President, members of the SEPG, Principal Consultants, Project Managers and software developers, and also administered a questionnaire to receive feedback from a larger number of respondents. The researcher observed the project management, knowledge and software process of the organisation, and the functioning of project teams within their work environment. Further, the researcher followed Miles and Huberman's (1994) suggestion that the analytic phase consists of three concurrent flows of activities that are data reduction, data display and conclusion drawing. This chapter presents how Yin's Case Study Protocol and Miles and Huberman's (1994) recommendation for analysis were followed and implemented.

### **7.1 Data Collection**

Data collection is concerned with providing sources of evidence, and a good research design must have multiple sources of evidence, Perry (2004) and Yin (2003). This research collected data through interviews, questionnaire and observation whilst conducting a twenty-six month case study at a large software development organisation. The interviews, questionnaire and observation were conducted in unison, thereby enabling the researcher to confirm key phrases, events, instances and insights from one method of data collection with the other method. For example, if the researcher observed an event or instance that he considered relevant to the research project, he would attempt to confirm that event or instance's relevance during subsequent interviews. Similarly, the researcher would attempt to confirm in subsequent observations what he considered were relevant events, issues and instances elicited during interviews conducted. During three particular extended site visits, the researcher was also able to request project managers to circulate a questionnaire, via e-mail, that required open-ended responses to a few chosen questions from the set of pre-

determined questions prepared by the researcher to help guide the interviews. The list of these pre-determined questions is provided at the end of the next section (7.1.1). This survey questionnaire was given primarily to obtain a wider section of response from across the organisation, and on certain occasions to clarify and reconfirm relevant issues and instances of previous face-to-face interviews, thus providing the research with multiple sources of evidence as recommended by Yin (2003). A sample survey questionnaire is attached in Appendix 5.

### **7.1.1 Interviews**

Silverman (2005) notes that interview data displays realities that are neither biased nor accurate, but simply 'real', while Kvale (1996) also advocates the interview method to seek and describe the meanings of central themes. Thirty-eight open-ended interviews were conducted with individuals within the organisation covering a cross-section of seniority that included an Executive Vice President, the second most senior person within the organisation, a Vice President, Consultants, Researchers, Project Managers, Project Leads, and members of the Software Engineering Process Group (SEPG). The depth in organisational hierarchy represented in the cross-section of individuals interviewed helped ensure that the interview data collected did not have an over-reliance on either easily accessible or elite respondents, as suggested by Miles and Huberman (1994). Table 2 below depicts the number of interviews and total number of interview-hours conducted, along with the organisational management designation of each individual. As can be seen from the table, a total of 62 interviews took place involving 38 individuals with varying levels of seniority and experience lasting a total of 100 hours.

**Table 2 Interviews**

<b>No</b>	<b>Designation</b>	<b>Number of Individuals Interviewed</b>	<b>Number of Interviews Conducted</b>	<b>Total Number of Interview Hours (Approximate)</b>
1	Executive Vice President	1	1	3
2	Vice President	1	4	5
3	Senior Researchers	2	2	4
4	SEPG Members	3	5	8
5	Principal Consultants	2	12	30
6	Group Leads	3	4	6
7	Project Managers	13	18	26
8	Project Leads	6	8	10
9	Developers	7	8	8
		<b>38</b>	<b>62</b>	<b>100</b>

The interviews were conducted on-site during five field visits to two different development centres and the organisational head office. All interviews were open-ended sessions that lasted more than an hour each. As suggested by Yin (2003), the researcher prepared a case study protocol that included a pre-determined set of questions based upon the theoretical propositions of the research that helped guide the initial part of the interviews. Yin (2003) states that “the protocol is major way of increasing the *reliability* of case study research and is intended to guide the investigator in carrying out the data collection from single case study.” The interviews were interactive and questions were asked to elicit each individuals experience and perspective about sharing knowledge effectively within software projects.

Two initial interviews, one with the Vice-President and another with the Executive Vice President, required the researcher to make an opening presentation about the research being conducted and its main aim and objectives. Thereafter, on receiving permission from the Executive Vice-President and Vice President, the researcher proceeded to ask questions and make notes of the interview. Sample interviews are attached in Appendix 4 of this report.

Below is the set of pre-determined questions, developed through the literature review, to help the researcher guide the interviews. The research hypothesis and questions presented a set of open issues to be examined that are addressed through these questions.

- Interviewee's name?
- How many projects has he/she worked on, i) previously and ii) currently?
- How many members in his/her team?
- What methodology (or software development life cycle) is being used for the current project(s)?
- What, in their opinion, are the important underlying constructs of the software development process?
- What was the primary vision for the organisation's knowledge initiative?
- How was the organisation managing its knowledge requirements before embarking on a formal knowledge programme?
- What is the initial task they perform when assigned to a project?
- Define tasks associated with the development process.
- During which stages of software development are important decisions (i.e. those where 20% decisions affect 80% of the project) required to be taken?
- What tools are used for such decision-making?
- What are the inputs and outputs for such decisions?
- Identify information and knowledge input and output?
- How are personnel assigned to specific roles?
- What are the specifications of the role requirements?
- How are knowledge attributes assigned to people?
- What are some of the challenges and opportunities offered by a new focus on knowledge in organisations?
- How do their team members benefit from their experience and knowledge of implementing previous software projects?
- What kind of organisational structure and culture best fit a new focus on knowing processes? How did your organisation develop an organisational knowledge culture?
- Does CMMI Level 5 certification ensure that relevant knowledge is being captured within the knowledge processes of the organisation, or does the organisation need to further specifically address its knowledge requirements?
- How do subsequent projects the interviewee's work on benefit from the experience and knowledge you gained while implementing previous/past projects?

- Which in their opinion provides a better perspective for developing software – project management or software engineering?
- How does knowledge flow within your organisation?
- What facilitates the flow of such knowledge?
- What issues/obstacles did the organisation face while implementing the knowledge initiative?
- Who within the organisation is able to understand the impact of the changes brought about by implementing the knowledge initiative?
- What changes has the knowledge initiative brought about in the organisation's business processes and work practices?
- How is knowledge shared across organisational and functional boundaries, or between specialists?
- How can the organisation learn more effectively from its environment?

### **7.1.2 Survey Questionnaire**

The organisation where the case study was conducted employs more than 108,000 individuals. Since it was not possible to conduct face-to-face interviews with a majority of the individuals, the survey questionnaire was administered primarily to get responses from a wider section of respondents. The questionnaire particularly benefited the research in the following instances:

- i. In a distributed project team, the researcher was able to get views and opinions of on-site team members, that is, those team members who were based on the client's site and were currently not available for interviews at the organisation's premises. Thus, the researcher was able to receive feedback and insights on a single project from distributed sources.
- ii. The questionnaire provided the research access to other distributed organisational employees that would not be possible for a face-to-face interview.
- iii. The questionnaire helped clarify and reconfirm relevant issues and instances of face-to-face interviews.

Table 3 below shows the number of questionnaires mailed and responses received along with the designation of the respondents. As can be seen from the table a total of 24 questionnaires were sent out of which 20 individuals responded. The researcher was subsequently able to communicate with the respondents via email to seek clarifications and



further discuss relevant issues related to the research, thus extending the questionnaire instrument into a more dynamic information gathering tool from remote subjects.

**Table 3 Questionnaire**

<b>No</b>	<b>Designation</b>	<b>Number of Questionnaires Sent</b>	<b>Number of Responses Received</b>
1	Project Manager	5	4
2	Project Lead	12	9
3	Developer	7	7
		<b>24</b>	<b>20</b>

### **7.1.3 Observation**

Observation provides valuable insights into the work processes and practices of an organisation. The researcher was able to observe the project management, knowledge and software processes of the organisation, and the functioning of project teams, during field visits while conducting the case study. Two ground rules were specified for observing team interactions and meetings to minimise the possibility that team members become conscious of the researcher's presence and thus not conduct themselves in a 'natural or normal' manner because of being 'observed':

- i) the researcher must spend a number of weeks observing team meetings and interactions, and
- ii) the researcher make a presentation to team members at the end of the session regarding the research and observations made, and answer any queries or clarifications

The first ground rule was carried out by ensuring that each field visit was for a minimum of three weeks. This was done to ensure that after the initial observation session, individual team members were familiar and comfortable with the researcher being present during such meetings and sessions. The second ground rule made team members aware of the research being conducted, and benefited the researcher by enabling him to conduct an inter-active group discussion, and also obtain confirmation and feedback about the observations at the end of each session. This played a part in strengthening the value and perceived importance of the trends that were observed and enabled early clarifications of

issues. The ground rules were established after discussion with the senior management of the organisation where the case study was being conducted.

The researcher observed meetings that included weekly reviews, design, project start-up, closure, and conference calls with on-site developers and clients, in addition to software development activities and daily team interactions. Weekly review meetings were on an average for approximately an hour and a half to about two hours, while project start-up, closure and design meetings were of considerably much larger length. Most design meetings lasted a minimum of a half day or four hour session, with some meetings lasting for three-quarters of a day to the complete day or approximately eight hours. Certain design meetings required to be resumed the next working day. Focused project start up meetings that involved initial stakeholders would typically last for half a day while the same was case the case for project closure meetings. However, preparation for these meetings includes providing the attendees with the details to be discussed prior to the meeting. Similarly, attendees at the quality review meetings were also provided with prior details regarding defects and their causes to be discussed. There was no typical length for quality review meetings.

The researcher was able to observe software processes and functioning of project teams within their work environment. This required the researcher to be present within the coded access areas of the team's workplace for a half day session at each instance. This enabled the researcher to observe, examine and make notes of team interaction, and work methods and practices. On many occasions the researcher was invited to sit in a Principal Consultant's room for an extended session, for example 8am to 2pm, and observe his functioning and also interaction with individuals who visited the room. Inevitably, the researcher would be asked to contribute to discussions and conversations on software processes, project management, issues on knowledge management and the organisation, with the Principal Consultant and individuals who visited. The researcher was able to interact in a similar manner with four other Project Managers. Table 4 below presents a summary of observations of the type of meetings and team interactions, along with average and total number of observation hours. The table shows that the researcher was able to observe 97 meetings, which were of 11 different types, and yielded 340 hours of observation. It is important to note that since the meetings were of different length, the average hours per meeting are approximate figures.

**Table 4 Summary of Observation Hours**

<b>No</b>	<b>Type of Meeting</b>	<b>Meeting Function</b>	<b>Number of Meetings Observed</b>	<b>Average Hours per Meeting</b>	<b>Total Number of Observation Hours</b>
1	Project Start-Up	Project Management	3	4 ½	13 ½
2	Design	Software Development	4	7 ½	30
3	Weekly Review	Review	18	2	36
4	Conference Calls with Clients and On-site Developers	Project Management and Development	5	Approximate 2 ¾ (One call was long = 4 ½ hrs)	18 ½
5	Software Development Activities	Software Development (Observation Time)	11	Approximate 3	33
6	Team Interactions	Project Management	15	2 ¾	36
7	Coding	Software Development	15	Approximate 2 ½	35
8	Quality Review	Quality (Software Development)	6	Approximate 3	18
9	Project Closure	Project Management	4	Approximate 10	40
10	Maintenance	Software Development	6	Approximate 3 ½	20
11	Interaction with Principal Consultant & Project Managers	Project Management	10	6	60
	<b>Total</b>		<b>97</b>		<b>340</b>

The first field visit provided the researcher with the opportunity to observe for one week. The subsequent field visits enabled the researcher to observe team interactions and meetings for periods of four, three, three and two weeks, respectively, thereby resulting in a total of thirteen complete weeks of observations over an extended period of twenty-six months. These observations were typically conducted from 8am to 6pm, Monday to Friday, at two of the biggest development centres of the organisation. This helped ensure that the researcher observed consistent day-to-day activities, and did not over-weight dramatic events as cautioned by Miles and Huberman (1994). Table 5 below depicts the observation hours conducted per week during field visits to the organisation. The researcher was able to spend a total of 340 hours of observations during 13 weeks of 5 field visits.

**Table 5 Observation Hours per Week**

<b>Field Visit</b>	<b>Number of Weeks</b>	<b>Total Number of Observation Hours</b>
1	1	20
2	4	110
3	3	90
4	3	80
5	2	40
<b>Total</b>	<b>13</b>	<b>340</b>

#### **7.1.4 Constraint during Data Collection**

A key constraint during data collection was the organisation's security measures, which did not permit an individual who is not part of the organisation to have any electronic storage or recording device on their person. This resulted in the researcher being unable to record, either audio or visually, any interview or observations. The researcher overcame this by following Spradley's (1979) suggestion that researchers keep four separate sets of notes:

- i) Short notes made at the time.
- ii) Expanded notes made as soon as possible after each field session.
- iii) A field work journal to record problems and ideas that arise during each stage of fieldwork.
- iv) A provisional running record of analysis and interpretation.

The researcher followed the above suggestions and maintained immediate short notes that highlighted key phrases and events to help recall the flow of discussion and interactions during interviews and observations. Immediately after the interview, the researcher would write down in detail what he was able to recall. Also, the researcher attempted to transcribe the interview on the same day that it was conducted. Similarly regarding observations, the researcher would immediately after the session write down as much as he could remember about the physical arrangement of the room and the interpersonal activities that had taken place, and also the key interactions and events. As and when required further interviews were scheduled to follow-up and seek clarifications or validate issues determined during previous interviews or observations.

### **7.1.5 Summary of Data Collection Activities**

The researcher was able to collect data through the three methods of open-ended interviews, questionnaire and observation within a large globally distributed software project organisation that currently employs more than 108000 individuals across 41 countries and implements mature software processes. These methods of data collection were conducted during 5 field visits to the organisation spanning a period of 26 months. The extended field visits enabled the researcher to be physically present at the organisation for a total of 13 weeks during which time he was able to interact with and interview individuals and observe team interactions, and also software development processes, and organisational work methods and practices. The researcher had access to observe 11 different types of team meeting and interactions that provided 97 instances of observation including the various software development and project management processes, and was able to conduct 62 open-ended interviews with 38 individuals ranging in seniority from an Executive Vice-President to software developers. These interactions, observations and interviews resulted in data collected from 100 hours of interviews, 20 responded questionnaires, and 340 hours of observations. The following section discusses the findings and how the data was analysed.

## **7.2 Findings**

As discussed in Section 7.1, the data collection process yielded 100 hours of interviews across a cross section of individuals, and 13 weeks of observation data over a period of 26 months at two main development centres of one of the largest software project organisations

in the world, which currently has more than 108,000 employees. The data collected and insights gained from the two centres seemed consistent with each other. However Miles and Huberman (1994) caution against quickly assuming insights as ‘typical’, and suggest safeguards to ensure the ‘representativeness’ of findings. This research followed the safeguards recommended and ensured that there was no holistic bias or over-reliance on plausibility and data elicited from elite informants or respondents.

The individuals interviewed provided their views and insights based upon their experience on projects implemented. The senior respondents were generally unable to recall the exact number of past projects they had worked on. This question was relevant as it was the experience they gained while implementing all projects, past and present, that they would be required to call upon for problem solving and decision making while implementing current projects. However, the researcher did ask the respondents the number of current projects they were working on. The answer to this question revealed that the respondents were providing views and insights that were influencing the implementation of 98 current projects. The Executive Vice President and Vice President and members of the SEPG were not considered directly involved or ‘hands on’ in the implementation of projects. In fact upon being questioned, a common reply was they had more “*strategic*” inputs in the implementation of projects. The Principal Consultants also viewed themselves in a similar vein, but were more involved in terms of reviewing different projects, advising improvements and problem solving. The Senior Researchers were involved and working on research projects as required by the organisation. Therefore Group Leads, Project Managers, Project Leads and Developers were involved in the active implementation of software development projects. This is displayed in Table 6 below.

**Table 6 Data Collected from Ongoing Projects**

<b>No</b>	<b>Designation</b>	<b>Number of Individuals Interviewed</b>	<b>Number of Projects Currently Implementing or Working on</b>
1	Group Leads	3	22
2	Project Managers	13	56
3	Project Leads	6	8
4	Developers	7	12
	<b>Total</b>	<b>26</b>	<b>98</b>

Each Project Manager or Project Lead mentioned in Table 4 was currently managing one or more software project, had been a software project manager or leader for at least one year, and had completed the implementation of at least one software project, when he/she was interviewed. These criteria ensured that the interviewees had a sufficient level of experience to offer insights into software project implementation and related processes. On the other hand, interviews with Developers were more focused towards their current tasks and activities, including the knowledge activities they were practising and the training they had received, either during induction or specifically for the project they were implementing. Thus the research was able to gain responses and insights into software project implementation at all levels that included the top strategic level, 'hands-on' management level, and development level, tasks and activities.

### **7.2.1 Data Reduction and Display**

Silverman (2005) states that "data analysis does not happen only after all the data has been safely gathered." The researcher was able to gain insights and make sense of the data while conducting interviews, discussions and observations. The large volume of data collected through the interviews, questionnaire and observation was examined and reviewed to ensure data accuracy. Where required, clarifications were sought to reconfirm the accuracy and relevance of key events, phrases and instances, thus beginning the process of checking and verification early in the analysis and conceptualisation stage of the research. The collected data was transcribed and coded, highlighting the relevant words, phrases, and events. A series of readings of the data helped assign specific codes to the pieces of text that represented important concepts and distinct responses during observations and interviews. Miles and Huberman (1994) state that 'displaying data in a systematic way has immense consequences for your understanding.' Miles and Huberman (1994) further state that 'displaying the data requires the researcher to think about the research questions and what portions of data are needed to answer them; it requires the researcher to make a full analysis, ignoring no relevant information; and it focuses and organises the information coherently.'

Following the above suggestion, the coded data was put into groups or categories, which resulted in twenty-nine groups, and adhering to the theoretical propositions and orientation of the research, attributes from within the data were assigned to each group, based upon their relevance to the group. For example, an initial category of knowledge flow was assigned attributes that facilitated or were directly or indirectly involved in the flow of

knowledge. The attributes were also derived from the data collected through interview responses and observations, such as social interaction, intranet portal or the organisation's monthly magazine.

The groups and their attributes were placed in a table to see "what's there", and this enabled the researcher to compare and identify similar attributes within the groups. The process highlighted data groups with certain similarities and themes which further made it possible to group the data in fewer groups or categories with consolidated attributes, which resulted in twelve categories. For example, training was initially considered a category by itself and had attributes such as specific roles assigned to it, was later itself assigned as an attribute to the knowledge and knowledge flow category where specific individual roles had already been assigned as attributes.

The groups were then placed in a column while the attributes assigned to each group were placed in rows within an adjacent column. This resulted in an initial two column by multiple row table or matrix that enabled the researcher to further compare and analyse similarities and relationships across categories and attributes, and form initial impressions. A third column was added to the table or matrix, and text explaining the first impressions regarding categories, attributes and their possible inter-relations, was added. This added clarity and was to later provide a chain of thought to the ongoing analysis and emerging insights, while providing a process of reformulation and ideas for further analysis, and reliability as suggested by Yin (2003). Continuing with the knowledge flow category example, Table 7 provides a sample of this process.



**Table 7 Data Reduction and Display**

No	Group/Category	Attributes	Comment (Text)
3	Knowledge and Knowledge Flows	Interaction – team members, customers	With whom – who are the individuals involved; where does this interaction take place; how does this interaction affect and influence project outcome?
		Reusability and lesser development time	Benefit of knowledge and its flow; can this benefit be measured; how is the organisation sure that this benefit exists?
		Process owners and Process Improvement Proposals	Who are the process owners – teams or individuals? What review and validity checks exist for such proposals?
		Lessons learned and best practices	How are these transferred and applied? Who is responsible for their integrity, validity and redundancy?
		Explicit and Tacit knowledge	Is there too much emphasis on explicit knowledge? What about dialogue and interactive problem solving?

The coding, groupings and assigning attributes were done by the researcher. NUDIST software was also used to display ‘group nodes’ and the data associated with them. However, Silverman (2005) notes that “using computer-assisted analysis of qualitative data is no substitute for thinking hard about the meaning of data”, in response to fears that computer technology will be used uncritically for data analysis. Critically evaluating the emerging themes from the data, and the subsequent groupings and their assigned attributes, enabled the researcher to interpret and gain further insights into the events and interactions

within the software project organisation. As suggested by Miles and Huberman (1994) the data was conceptualised through a mapping process whereby themes were identified and related. The categories formed through this process were further examined to assign attributes that enable the flow of knowledge within the development projects of the software organisation. Therefore, following Miles and Huberman (1994) suggestion, the following four criteria were adopted while determining and assigning attributes to each category.

- i) Each attribute must be mentioned and supported by at least two respondents during the interviews.
- ii) Respondents should have provided instances of how a particular attribute influenced their work methods
- iii) Each attribute should have significant relevance within the literature to its assigned category, and thereby adhere to the theoretical propositions of the research, or should offer new insight into the research because of its relevance
- iv) The interview data supporting each attribute is supported in unison by the data collected through observation, and vice-versa.

The twelve categories and their attributes are presented in Table 8 below:

**Table 8 Categories and Attributes**

No	Group/Category	Attributes (based upon emerging themes)
C1	Interaction	A1. Amongst team members A2. With on-site teams A3. With customers A4. With senior management A5. Formal/informal A6. Coordination A7. Collaboration
C2	Human Resources (HR) and Teams	A1. Emphasis on training – focused and planned for roles A2. Competencies A3. Feedback/appraisal A4. Top Management Support A5. Functional/Technical Experts A6. Mentoring/role models A7. Leadership

No	Group/Category	Attributes (based upon emerging themes)
		<ul style="list-style-type: none"> <li>A8. Manpower Allocation Task Committee (MATC)</li> <li>A9. Team Track (Tool)</li> <li>A10. Continuous learning</li> <li>A11. Communication</li> <li>A12. Motivation and Commitment</li> <li>A13. Cross functional/Virtual</li> <li>A14. Location – collocated/on-site</li> <li>A15. Skill sets/competencies</li> <li>A16. Roles and responsibilities</li> <li>A17. Structure/selection</li> <li>A18. Team building/bonding</li> <li>A19. Trust – especially localised</li> </ul>
C3	Knowledge and Knowledge Flows	<ul style="list-style-type: none"> <li>A1. Initiative</li> <li>A2. Review meetings/comments</li> <li>A3. Reusability and lesser development time</li> <li>A4. Shorter learning curve</li> <li>A5. Process owners and process improvement proposals</li> <li>A6. Knowledge Vision</li> <li>A7. Knowledge Strategy</li> <li>A8. Explicit knowledge helps at the micro level while tacit knowledge helps at micro and macro levels</li> <li>A9. Documents – checklists/templates</li> <li>A10. Peer Reviews</li> <li>A11. Thinking sessions/ Brainstorming</li> <li>A12. Knowledge Transfer</li> <li>A13. Infrastructure</li> <li>A14. Body of Knowledge (BoK)</li> <li>A15. Benchmarking</li> <li>A16. Portal/Assets Library/bulletin boards</li> <li>A17. Mail corner</li> <li>A18. Monthly magazine</li> </ul>

No	Group/Category	Attributes (based upon emerging themes)
		<p>A19. Software Engineering Process Group (SEPG)</p> <p>A20. Lessons learned and best practices</p> <p>A21. Capability</p> <p>A22. Scalability</p>
C4	Decision Making	<p>A1. Very grey process</p> <p>A2. Past experience</p> <p>A3. People capability changes</p> <p>A4. New technology and tools develop</p> <p>A5. Planning, estimation, resources, budgeting, reviews</p> <p>A6. Requirements, design and testing phases</p> <p>A7. Change requests</p> <p>A8. Risk Analysis</p>
C5	Project Start-up And Closure	<p>A1. Start up meeting to define ground rules</p> <p>A2. Gather right experienced people (team selection) and technology</p> <p>A3. Refer to the portal for best practices, metrics, project plan, schedule and similar project profile code</p> <p>A4. Integrated Project Management System (IPMS)</p> <p>A5. Thorough analysis</p> <p>A6. Scoping</p> <p>A7. Quality management plan</p> <p>A8. Identify stakeholders</p> <p>A9. Contacts/proposals</p> <p>A10. After Action Reviews</p> <p>A11. Project Closure Analysis/Report</p> <p>A12. Retrospectives</p> <p>A13. Monitoring and Control</p>
C6	Process Constructs	<p>A1. Very clear requirements</p> <p>A2. Skill sets – proficiency/training</p> <p>A3. Good testing</p>

No	Group/Category	Attributes (based upon emerging themes)
		<ul style="list-style-type: none"> <li>A4. Well defined tool based processes with defined roles and responsibilities</li> <li>A5. Balanced teams – composition and motivation</li> <li>A6. Good analysis and design</li> <li>A7. Risk Mitigation</li> <li>A8. Maintenance – good documentation and knowledge base</li> <li>A9. Quality management system and reviews</li> <li>A10. Methodology</li> <li>A11. Best practices</li> <li>A12. Process discipline</li> </ul>
C7	Experience	<ul style="list-style-type: none"> <li>A1. Formal routines – greater the experience, lesser use of formal routines, and control and discipline</li> <li>A2. Repositories – Case based reasoning (CBR)</li> <li>A3. Using technology such as portal or cartography, and involving users to participate in its development</li> <li>A4. Social activity – more flexible and adaptable</li> <li>A5. Cooperative work</li> <li>A6. Story telling</li> <li>A7. Lessons learned cannot be easily transferred from one setting to another</li> <li>A8. Expert opinion</li> </ul>
C8	Quality	<ul style="list-style-type: none"> <li>A1. Defect prevention</li> <li>A2. Considered an excellent area for learning, and especially for training new employees</li> <li>A3. Reflection</li> <li>A4. Quality Management System (QMS)</li> <li>A5. Review requirements, code and test</li> </ul>

No	Group/Category	Attributes (based upon emerging themes)
		<p>cases</p> <ul style="list-style-type: none"> <li>A6. Causal Analysis and log into IPMS</li> <li>A7. Defect prevention meeting reviews</li> <li>A8. Revision of checklists, templates and coding standards</li> <li>A9. Delivery checklists</li> <li>A10. Best learning published in portal/assets library</li> <li>A11. Branch level testing</li> <li>A12. Increased reliability and performance of the product</li> <li>A13. Log defects in the QMS</li> <li>A14. New inductees to train on causal analysis</li> </ul>
C9	Globally Distributed Organisation	<ul style="list-style-type: none"> <li>A1. Formal Knowledge sharing</li> <li>A2. Loss of richness of knowledge with less face-to-face interaction</li> <li>A3. Regional knowledge sharing</li> <li>A4. Loss of 'teamness'</li> <li>A5. Cultural differences amongst globally distributed staff and work practices</li> <li>A6. Local acculturated knowledge champions</li> <li>A7. More frequent regional interaction (through individual visits)</li> </ul>
C10	Culture	<ul style="list-style-type: none"> <li>A1. In formal, quasi-formal and formal</li> <li>A2. Team/departmental silos – No concept of Boundary Brokering</li> <li>A3. Reward systems</li> <li>A4. Communities of Practice</li> <li>A5. (Now) Knowledge Champions</li> <li>A6. Individual developers learn from peers rather than refer to a Process Guide</li> <li>A7. Bench strength</li> <li>A8. Absorptive Capacity</li> </ul>

No	Group/Category	Attributes (based upon emerging themes)
		A9. Centres of Excellence
C11	Processes	A1. Ongoing, day-to-day activities A2. Existing systems A3. Properties and capabilities A4. Iteration A5. Continuous and repetitive manner A6. Assess and improve – constantly changed and refined A7. Knowledge intensive A8. Innovation, creativity and autonomy A9. Knowledge flows not clear in advance but evolve during the process A10. Integrity of knowledge – its origin and application A11. Socialisation and informal exchange A12. Constant documentation A13. Reuse of processes as patterns A14. Discipline and control versus creativity and flexibility
C12	Feedback	A1. Appraisal A2. Process A3. Learning A4. Defects A5. Rapid A6. Mechanisms A7. Weekly A8. Lessons learned A9. Need to develop measures A10. Customer Satisfaction Index

### 7.2.2 Analysis and Conclusion Drawing

An evaluation and analysis of the above categories and their attributes based upon the emerging themes, combined with the insights gained while conducting the case study.

provide a rich empirical basis to analyse and present the flow of knowledge while implementing projects within a software project organisation. This research primarily followed the first and “most preferred” strategy of ‘relying upon theoretical propositions’ recommended by Yin (2003) and Miles and Huberman (1994). Further, developing categories and their attributes, and comparing them iteratively for similarities, interactions and relationships, could be considered a form of pattern matching as also suggested by Yin (2003) and Miles and Huberman (1994). Yin (2003) further states that explanation building, time-series analysis, and logic models are also forms of pattern matching analysis. Miles and Huberman (1994) suggest that data analysis consists of three activity flows of data reduction, data display and conclusion drawing and verification. Having performed data reduction and display, the research proceeded with the activity of conclusion drawing which Miles and Huberman (1994) describe as ‘beginning to decide what things mean, noting regularities, patterns, explanations, possible configurations, causal flows and propositions.’ This process of evaluation and analysis is presented in the following chapters.

### **7.3 Conclusion**

This chapter discussed the methods of data collection adopted by this research, and the findings. The emerging themes from the transcribed and coded data were categorised and attributes assigned based upon qualifying criteria. The chapter described how the data was collected, and the rationale and method adopted to code, categorise and display this data. These findings provide an empirical base for further discussion, evaluation and analysis presented in the next section.



Part III Evaluation, Analysis and KM  
Capability Framework

## Chapter 8 Evaluation and Sense-making

Following Miles and Huberman's (1994) recommendation towards analysis and conclusion drawing, this chapter discusses the evaluation and insights gained by analysing the data collected while conducting the case study at organisation XYZ. The evaluation and discussion provide insights into the organisation's knowledge vision, strategy, initiative, flow and issues addressed while developing the knowledge management initiative. During the evaluation, an emphasis was put on identifying the flow and exchange of tacit knowledge as discussed in Sections 2.5 and 4.3, and the particular issues of developing a knowledge initiative in a distributed environment. The evaluation and discussion emerge from the categories and attributes as they appear in Table 8, Chapter 7. Each category and attribute is linked in the discussion and in the interest of readability only its first appearance is marked in brackets. For example Knowledge Strategy is an attribute that appears in Category 3 of Table 8, and is therefore marked (C3-A7).

### 8.1 Knowledge Vision

The primary vision for the knowledge initiative (C3-A1) at XYZ is to ensure that individuals can rely upon having the right knowledge available to them at the right time. In order to leverage learning while developing software in project organisations, the top management was of the view that the entire organisational knowledge should be consolidated and that individuals interacting with the organisation's clients have complete access to this knowledge. The effort of consolidating knowledge was expected to minimise the effort dissipated in redoing learning that had already happened elsewhere. Knowledge vision of XYZ appears in Category 3 and is Attribute 6 (C3-A6) of the Categories and Attributes Table 8.

The interviews and observations revealed that the key drivers of the KM initiative at XYZ were better quality, better productivity due to artefact reuse, cycle time reduction, virtual teamwork, greater market awareness, higher revenue growth, increased customer satisfaction, reduced risk brought about by diversifying into new technologies, domains, geographical areas, and services. As stated by a Group Lead that *"we at XYZ are always looking towards diversifying into new technologies, domains, products and services, and*

*geographical areas, and therefore need the constant backing of our organisation's collective knowledge and capabilities."*

XYZ had many previous systems. Initially there was the paper-based system which involved established processes of submitting project reports, information and empirical data. These systems eventually became islands within the office automation systems. In 1992-93 XYZ had adopted the ISO 9000 model and most reports were in a fairly established format. These reports formed the basis of XYZ's asset library. Senior management at XYZ is of the view that the organisation's knowledge initiative evolved rather than started due a recent knowledge vision. Also XYZ is currently into its "*fourth generation system*" and the managers state that their biggest challenge is scalability (C3-A22) of knowledge, that is, from the initial knowledge requirements of 10000 employees, the system now has to cater to the knowledge needs of more than 108,000 employees with a potential for further growth. Therefore the senior management view is that XYZ now requires and needs a system that is independent of people, and that is what their latest knowledge effort has tried to address. A brief description of how the knowledge initiative at XYZ evolved and its key features are discussed below.

## **8.2 XYZ's Body of knowledge**

The system of sharing technical and experiential knowledge among peers existed at XYZ for a long time through the use of email and intranet. Initially, after the physical document library, the organisation started urging individuals to document their experience in the form of MS Word documents to enable them to be collected and shared across the whole organisation. Such documents were initially kept as soft copies in a server that was accessible to all individuals within the organisation. This repository of experiential knowledge was called 'Body of Knowledge' (BoK), and appears in Category 3 and is Attribute 14, (C3-A14) Table 8.

The BoK entries were classified to make them easy to use, and the categories included both technical and non-technical issues and subjects, in an attempt to ensure that the entries were useful to all departments of the organisation. Individuals were required to fill a form and submit the entry in both soft and hard copies, along with the entry's categorisation. While the soft copy was placed on the organisation's server, the hard copy was stored in the library for easy reference.

The above system was sufficient when the organisation was smaller and had less than 1000 employees. As the organisation expanded in size and distributed geographical locations, the need to make this BoK accessible to all organisational individuals became even more critical. Thus the organisation moved towards a web-based system in the mid-nineties. The entries were submitted via email and stored within the intranet after the system was upgraded to accept web-based submissions. The web-page contained a write-up about the concept of BoK, and the categories for submission. The contributors were able to download a template that contained an entry-form, attach the template to the entry, and submit the entry via e-mail. An online entry form was created when it was observed that contributors typically missed certain information critical to describing and classifying the entry. The online entry form, while enabling online web-based submission, ensured that all the relevant and required information was collected and posted, further improving the search facility on the BoK and increasing the use of the system. A committee reviewed the entries, for readability, correctness, and intellectual property rights issues, before they were included in the system.

Initially, as mentioned previously, individuals were requested to document their learning and experiences. Software developers who believed in their ability to execute tasks rather than document their experiences had to be convinced that their experience was invaluable to the organisation and its employees. An awareness programme was conducted for individuals within the organisation, and apart from an incentive scheme, contributions to the knowledge management system were included as an item in the organisation's performance appraisal (C2-A1) system. Thus it was important for the organisation to create awareness for the KM initiative to be developed. This is a significant observation, and the research later establishes that organisations need to communicate their knowledge vision and translate it into values during the initial stages of developing a KM initiative, and introduce the KM initiative as a part of induction training during the later stages of development.

### **8.3 Knowledge Strategy**

One of the main ways a software organisation captures and re-uses knowledge is through process definition as guidelines, templates, checklists and process assets evolve and are documented. At XYZ, modules such as the Integrated Project Management System (C5-A4), Quality Management System (C8-A4), and the Body of Knowledge (C3-A14) are

available on the intranet, while other successful experiences and learning are disseminated through seminars and best-practice sessions within the organisation.

XYZ's knowledge strategy (C3-A7) is hybrid as it draws upon both the personalisation and codification strategies, and has evolved through knowledge usage over many years. XYZ has developed a knowledge culture at the local, branch, regional and corporate levels. In a globally distributed organisational environment, each local centre at XYZ since late 2006 has a knowledge champion who is well versed in the local culture. The local knowledge champions interact with branch knowledge managers, who in turn interact at the regional and corporate levels leading to an overall knowledge manager. The overall corporate knowledge is managed and disseminated by the SEPG (C3-A19). The intranet portal is the primary tool for collating and disseminating corporate knowledge.

Top management support is required for any KM initiative to succeed, and at XYZ such support is available. The scope of activities in encouraging the KM initiative extends from imparting learning to consolidating organisational knowledge. Thus the SEPG provides leadership to the effort, reviews content and progress of the initiative, ensures that goals set are realistic and implementable, and advises the top management about the support required for the KM initiative in terms of resources such as people, budget and other infrastructure. Currently at XYZ, the SEPG plays a facilitator's role with an emphasis on decision-making, project management, and quality. The SEPG meets every month where it reviews successful and unsuccessful projects, submits and collates reports on best practices and lessons learned, and disseminates the knowledge gained. The SEPG also maintains the assets library, publishes documents and check lists, certifies tools and promotes artefact reuse.

In addition to the above, the SEPG is involved in project start-up meetings where it studies the feasibility of project, sets the standard for quality and good practices, promotes the sharing of previous experience and identifies the possible pitfalls that maybe encountered. It interacts with the Man-power Allocation Committee regarding the availability of personnel and skills, and separately also ensures that technology is available to implement the project. Most importantly, the SEPG develops the implementation strategy for the organisation, identifies areas of improvements, develops training programmes, maintains the knowledge repository, reviews lessons learned and best practices, shares experience and provides a cross project perspective that benefits current and subsequent projects.

At the local development centres and levels, working groups evolve over time to facilitate implementation of the KM initiative to the individual level. These working groups include knowledge practice champions who drive the KM initiative within their respective areas of influence. As individuals realise the benefit of accessing and using the KM initiative they popularise it amongst their peers. Further, since XYZ has employed a majority of its personnel after implementing the knowledge strategy, and fifty percent of employees have been at XYZ for less than 3 years, the current knowledge culture is an integral part of their induction and training programmes. XYZ hires ‘bench strength’ that provides a bigger talent pool, ongoing training, and possibilities of job-rotation, who are also made familiar with the KM initiative.

#### **8.4 The Knowledge Initiative**

The knowledge initiative at XYZ evolved with the objective to provide the ‘right knowledge to the right person at the right time’ and thereby improve decision-making. As the work environment became more dynamic, uncertain, volatile and competitive, employees required a knowledge base upon which to make the ‘right decision at the right time’ to cater to customer needs and requirements. Customer satisfaction at XYZ required managing knowledge of the organisation’s clients’ business domain, applications and networking.

In understanding the organisation’s clients’ business domain, there was a need to make a database of the organisation’s clients, the nature of their businesses and operations, and the various characteristics and features for the software developed to achieve and deliver. There was also a need to understand the specific technology preferences of each client and the type of service they further provide their customers, and what value was needed to be added to that particular service. The knowledge initiative was expected to enrich the interaction with the client and in the process also enhance the value judgments and decision-making of the individuals involved, both at the client organisations and at XYZ, thus motivating them to learn from their experience.

Technically, the organisation’s individuals require knowledge about the software being developed and the technology to be used to develop the software. However, individuals within XYZ also require knowledge about the organisation and its objectives, roadmaps, current performance, future plans, vision, mission, values and virtues, functioning of departments, teams, groups, budgets, plans, training, skills and competencies. Therefore,

while the management at XYZ was confident about the knowledge generated and captured within, the knowledge initiative required addressing the issues involved in using knowledge captured outside the organisation, such as who developed the knowledge or integrity of knowledge, who experiences the changes taking place, who is able to understand the impact of the changes brought about by the change in knowledge, who needs to acquire the knowledge, what is the future use for the acquired knowledge, and what are the strategic reasons for acquiring the knowledge.

#### **8.4.1 Issues addressed while implementing the knowledge initiative**

While the KM initiative evolved over a period of time, it involved the consolidation of all knowledge assets, that is, it went through a consolidation phase. The planned integration with the Integrated Project Management System (IPMS) (C5-A4), where the entire project life-cycle is tracked, and artefacts are extracted, is an example of the evolution and consolidation of how the KMS became an integral part of the organisation's delivery and support systems.

The challenges to introducing the KM initiative included promoting a sharing culture, building and sustaining momentum, deploying IT infrastructure, and ensuring quality and currency of content. Steps taken to address these challenges included identifying projects that demonstrated significant knowledge use, studying the relevant characteristics, features, and competitive advantage of these projects such as reduction in project life-cycle time, customer satisfaction (C12-A10), lower cost, increasing competencies (C2-A2), shorter learning curves (C3-A4), and reduction in response time to requests for information and proposals, and publicising the benefits within the organisation.

Training programmes and a reward system (C10-A3) were implemented to encourage participation, and goals set with regard to knowledge contribution. A column was included in individual's weekly activity report regarding the proportion of time spent on KM activities, and in certain cases responsibility was mandated for implementing sustainable knowledge capture and reuse strategies. Individuals were educated on the features and benefits of the knowledge initiatives through innovative publicity material such as posters, mouse pads and coffee mugs with the KM logo.

Towards the later stages a major challenge was the sheer scale of the initiative as the knowledge initiative had to cater to the knowledge needs of so many existing employees.

The knowledge initiative had to ensure that individuals could rely upon having the right knowledge available to them at the right time. Therefore, individuals had to trust the system and the knowledge available. Such trust was instilled by ensuring the relevance and quality of content in the KM system. New employees joining the organisation were made familiar with the knowledge practices as a part of their induction programme. However, existing employees had to be made aware of the KMS and encouraged to use it, by means of conducting seminars, training programmes and workshops, and through articles and prominent displays in the organisation's monthly magazine. The knowledge initiative was further integrated in the organisation's work practices and processes through best practice sessions, project retrospectives, online expert moderation, bulletin boards, and notice boards.

## **8.5 Knowledge flow within XYZ**

Knowledge flows (C3) in XYZ as a part of the daily activities of individuals and team members. The senior management view is that organisational knowledge flows in all directions, and there is no particular model flow as XYZ is a vibrant organisation and knowledge flows across hierarchies and departments in a quasi structure. They consider the organisation to be a mix of informal, quasi-formal and formal (C1-A5) culture, with knowledge moving across all levels through formal and informal channels. This view is apparent in quotes from the interviews conducted such as:

*“Knowledge within our organisation, flows bottom-up with regard to technology, top-down with regard to policies and strategies, and also laterally in a chaotic, flexible manner without constraints.”*

*“The medium for knowledge to flow is within our tools, email, notice boards, magazines and mail corner.”*

*“Also, our organisation has interaction at all levels. This interaction could be either formal or informal, and in a structured manner. Unstructured interaction does exist, but cannot be accounted for.”*

A direct question to Project Managers and Project Leads at XYZ resulted in the following being identified as the main channels of knowledge flow within the organisation:

intranet portal (C3-A16), mail corner (C3-A17), monthly magazine (C3-A18), technology support and associated tools, Asset Library (C3-A16), knowledge sharing sessions also known as expertise sharing, training plan for each month which includes classroom and computer based learning, mandatory training on tools such as Integrated



Project Management System (IPMS) (C5-A4), Internal Knowledge Management System (IKMS), defect prevention (C8-A1), causal analysis (C8-A6), customer interaction (C1-A3), reviews (C3-A2) including peer reviews (C3-A10), e-mail, meetings (C3-A2), mentoring (C2-A6), personal interaction and communication (C2-A11), telephone and teleconferencing.

Since late in 2006 the organisational structure has a Chief Knowledge Officer (CKO) who is a member of the SEPG, who in turn are the owners of the knowledge processes thereby reflecting top management support and commitment to a knowledge culture. Knowledge Champions report to the CKO and there is a whole hierarchy of people who ensure that knowledge flows in a process-driven manner through the organisation. The size and scale of the organisation's knowledge structure now requires that a formal structure is in place to ensure that the knowledge needs are met, because relying only upon informal channels would mean that the knowledge exchange is carried out on an ad-hoc basis. However, initiatives such as an open plan office layout provide individuals and teams with an opportunity to have frequent meetings, both in a formal and informal setting.

An important aspect of informal and formal (C1-A5) knowledge exchange at XYZ is individual visits and interactions. For example, when a Delhi-based employee visits the Sydney or London office of XYZ, he or she interacts with XYZ employees at that location, both formally and informally. XYZ views these visits as central to their knowledge exchange requirements. At a more formal level, a recent development is that local knowledge champions travel to formal knowledge exchange meetings which range from the local, branch or regional, national and corporate levels.

Other formal channels of knowledge exchange currently at XYZ are Communities of Practice (C10-A4) and Centres of Excellence (C10-A9). Such communities are both local and virtual and can be created within the portal, for example the Architects Forum. The organisation also conduct's technical seminars, and weekly meetings, as well as learning weeks and library weeks for its employees.

E-mail is the primary mode of communication within the organisation while teleconferencing is used to communicate with clients. An important channel of knowledge flow within XYZ is the intranet portal which is considered a frequent mode of knowledge exchange, communication and coordination within interdepartmental teams. Significant policies and important organisational decisions are shared over the intranet, as are lessons

learned. The organisational vision, mission and values are available as screensavers for organisational computers. Employees are made aware of best practices and lessons learnt through success stories and presentations on the intranet portal. Also shared on the intranet is information related to changes regarding regulations, customers, budgets, plans, taxes and their impact on the organisation.

Below are several quotes extracted from the interviews conducted to highlight the importance accorded and reliance upon the intranet portal within the organisation:

*“ABC-portal is XYZ”.*

*“ABC-portal is the lifeline of XYZ.”*

*“ABC-portal is the lifeline of our knowledge needs.”*

*“The intranet portal provides 24 hour access to resources that employees at XYZ would require.”*

*“ABC-portal helps with process changes, defect prevention, and technology changes.” “ABC-portal also helps in process, estimation and white paper differentiation.”*

As highlighted in the quotes above, ABC-portal (C3-A16) is considered the life-line of knowledge and information flow within XYZ. ABC-portal deployment has increased organisation-wide knowledge management processes. The portal enables several individuals in different places and periods of time to work on common projects, and modify established organisational work methods. Senior management modified organisational work methods by facilitating teamwork amongst task-oriented groups of people as compared to a previous organisational culture of departments. However, it was observed that a departmental silo effect still remained in the day-to-day activities of the organisation, though the portal was expected to help overcome this effect with a greater emphasis on facilitating knowledge exchange through any-time, any-place communication, collaboration and coordination. In addition to sharing knowledge, expertise and creating organisational memory, the portal was also considered to help reduce costs, increase productivity and enhance satisfaction and efficiency.

While developing ABC-portal, XYZ needed to make critical judgments upon how to conceptualise knowledge practice and whose representation of knowledge to rely upon. It was only through ongoing use and redesign that XYZ has been able to design the portal to meet its knowledge requirements. Ongoing use also helped make clear the conceptualisation of XYZ's work and knowledge practices and requirements. XYZ's work

methods and knowledge processes involve organisational context, change of work practices, technology change and acceptance by individuals.

The portal provides individuals with access to the central repository that includes the various templates, checklists, guidelines, process assets and Project Assets Library (PAL) (C3-A16), BoK, Process Improvement Proposals (PIPs) (C3-A5), Lotus Notes and collaborative (C1-A7) tools such as Infinity and Interpreter, discussion and bulletin boards (C3-A16), instant messenger and email, decision analysis tools, IPMS, Customer Relationship Management System (CRM), human resources management system (HR), and the Electronic Knowledge Management System (EKMS). This version of the EKMS has now been replaced by DEF (C3-A16) which is a new structured KMS that consolidates all scattered assets into one system that caters to global needs of 108,000 diverse employees. DEF retains the features of central repository but is a value addition to the existing search and retrieval features of the previous system. The interviews revealed that users were involved in the design and evolution of DEF, and therefore it is being used even more frequently than the previous EKMS.

New features are constantly added to the portal. An example of such an addition is the feature of pattern analysis that helps in identifying patterns for solutions to recurring problems by picking out best practices and lessons learnt. Apart from knowledge or artefact reuse, patterns help identify the source of problems revealed after customers use the software product. Customer complaints are logged into the central repository which is accessed via the portal in a systematic manner. After the specific problem is sorted out, the logs are analysed over a longer period for patterns. The effort is to identify the root causes of these problems and then proactively prevent their recurrence. These patterns are shared both within the organisation and with customers. In doing so, XYZ is able to advise customers about potential problems and suggest appropriate solutions. This pattern seeking in customer logs also helps the organisation understand their customers' future requirements better, and translate this understanding to other customers with similar requirements. Thus patterns are an indicator of hidden process potentials and point out opportunities for an alternative process design and further improvements.

#### **8.4.2 Process Assets Library**

Process assets are specific assets generated for a particular project, but stored for use by other projects. These assets were created when the need and importance of storing specific

reusable artefacts from specific projects was realised. Due to legal requirements also, the organisation stores artefacts of all projects implemented. These re-usable process assets are stored in a repository to minimise duplication of effort in creating similar artefacts for various projects. The intranet enables individuals to submit process assets generated by them, and also search, download and use existing process assets. Such submissions can be made while executing a project or upon its completion. Typically, the process assets are artefacts or outputs from a project, and any new learning or knowledge acquired and lessons learned that might be useful to other projects, are entered in the BoK system. These entries in the BoK consist of articles written by individuals based upon their experiences while implementing projects. Questions during the interviews revealed that typical examples of process assets relate to project management plan, configuration management plan, schedule, standards, checklists, guidelines, templates, training material, and developed tools and their related notes.

The project group associated with the project verifies the entry before adding it to the repository. The project group also ensures the ownership of these process assets, which also include process improvement proposals. The assets are archived based on the type of processes used in the project, for example, development process, maintenance process, or re-engineering process. This categorisation helps in simplifying the search and retrieval of relevant assets. The assets are periodically reviewed and relevant artefacts are included in the organisational documents, which are made available to other projects being implemented within the organisation.

### **8.4.3 Tacit Knowledge Exchange and Flow**

CMMI Level 5 accreditation at XYZ includes features and key process areas of the People's Capability Maturity Model (CMMI), where the people aspect is integrated into the processes and includes the people knowledge requirements for the process. Learning programmes for newly recruited employees address the CMMI Level 5 requirements. Training (C2-A1) is planned for roles, and in certain training programmes project leads and team members are merged. Training is mandatory and the emphasis during training is on processes as required during software development. Personal development programmes for each individual are devised according to their skills and proficiency level to ensure a certain standard.

The organisation lays an emphasis on training. The training input is technical as well as functional and relates to both technology and management. The system of competency

profiling helps identify specific training needs for all employees, matching their skills and performance. The organisation's philosophy is to provide training that is need based, focused, regular, measurable, and quality driven.

The management philosophy is to empower individuals with a sense of ownership regarding the activities they perform and tasks executed. The management considers empowerment to mean providing authority, responsibility, resources and knowledge within the 'right environment to make the right decisions at the right time'. The organisation functions with self-managed teams, but within an explicit system of roles, skills, and mapped processes as defined by CMM Level 5.

Project managers are expected to lead and mentor (C2-A6) their teams, while inculcating trust (C2-A19), faith, honesty, transparency, and integrity as values within the team environment. The team members are expected to be given the freedom to function within a working framework, where informal sharing of knowledge, face-to-face interactions, and open doors are significant features of the informal culture of the framework. The researcher observed the accessibility and availability of senior managers for decision-making and problem solving. Specific instances included immediate 'pop-in' discussions with consultants and group leads. The top management of the organisation has remained the same for a relatively long time signifying the low attrition rate of employees in the organisation. In fact, XYZ has the lowest attrition rate amongst all competitor and peer organisations. This continuity has helped in developing the informal and self-fulfilling organisational culture that motivates individuals to create, contribute and share knowledge.

Tacit knowledge exchange and flow at XYZ was observed to be facilitated through interaction, both formal and informal. Daily, weekly and monthly meetings are conducted and their minutes circulated. In weekly meetings amongst 36 team members, 12 Project Leads (PL) and one Project Manager (PM) working on *different* projects for the same customer, it was observed that the PM would have different individuals chair the meetings and write the minutes each week. On direct questioning the PM asserted that this ensured that all individuals were aware of the status, progress and problems of all projects related to that particular customer. This awareness was particularly important in solving recurring problems. All individuals were asked about particular problems they were facing in executing their tasks. The PM, while drawing upon his experience, advised them with suggestions to problem-solving, and recalling how similar problems had been solved while implementing previous tasks or projects. The PM also advises the team-members how to

handle certain stakeholder issues, and updates his team members on the status of the projects. Thereafter, the PM invites agenda points for next weeks meeting. and suggests that the team have another meeting within the week that he terms as a ‘thinking session to enhance delivery’, and for which he suggests that a Senior Consultant is invited to be the Moderator.

These regular meetings clearly depicted the features and characteristics of mentoring, role models (C2-A6), brainstorming (C3-A11), feedback (C12), learning by doing, hand holding, open work environment, team problem solving, and knowledge exchange at the organisation. A significant observation was that individuals seemed to prefer to learn from peers and would rather ask a colleague than refer to a manual or process guide. Also, after a period of learning, the project manager was of the view that individuals need to apply the learning by taking on more responsibility, problem solving and decision making. Interview quotes depicting the leadership qualities of mentorship, and delegating responsibility and authority by project managers are:

*“I view my role as a PM to be a role model and leader. I aim to benefit my team members through mentoring, as I feel that class room training does not help. Also, I delegate work with authority, and thus aim to make my team members more responsible and train them in decision-making.”*

*“I encourage my team members to observe while working with others as the most important learning is through observation. I observe deficiencies within my team members and aim to strengthen their weaker areas.”*

An important suggestion that emerged from the discussions during the interviews was to *“employ trainees in Quality Assurance before coding. This can be very beneficial as they do the causal analysis and understand what went wrong.”*

The organisation promotes learning from clients and users as well as learning from peers and literature. Roles and skills (C2-A16) are defined for positions depending upon competency profiles and process requirements. This definition helps in effective recruitment and subsequent training. The competency profile also forms the basis of performance measurement, and plays a part in designing incentive structures. The incentive structure is based upon individual’s roles, skills and competencies and their

individual and team performance. This transparent system of performance management motivates individuals to fulfil their task commitments and self-development.

Individuals are allowed flexible working hours as long they put in the minimum hours per day as per their working contract, and are available between 10 am to 4 pm for any impromptu meeting which they might be required to attend. The organisation values individual's satisfaction and employee relationship surveys are conducted periodically to assess satisfaction with work related benefits, and communication within the organisation. Learning from leaving employees is also highly valued and exit interviews focus on the reasons for joining the organisation, whether those objectives were fulfilled, and reasons for leaving. Data collected from exit interviews are analysed and shared with the objective of creating new initiatives and taking corrective measures to reinforce interaction amongst peers across functional groups.

#### **8.4.4 The Knowledge Map for People (KM-P)**

The KM-P is an organisation-wide directory, in the form of 'Yellow Pages', for locating expertise (C7-A8) within the organisation. This appears to conform to the cartographic school (C7-A3) in Earl's (2001) taxonomy of knowledge management strategies. The system is designed in a multi-level hierarchical structure, with the top-level containing topics like project management, processes, technology, application domain and culture, with the deeper levels containing more specific details. The system is envisaged to allow the knowledge user to locate a knowledge provider, and make the tacit knowledge available with an individual available to the entire organisation. The system initially started with voluntary registration with consenting individuals making themselves available for enquiries on certain topics from others. However, over time there has been an increasing reluctance to be labelled as an expert, as observed by Desouza (2003).

The growth in this phenomenon meant that the organisation was unable to fully utilise its knowledge resources. Following a number of discussions, the organisation, acting upon recommendation by the researcher, linked registration with the individual's expertise, skill and competence levels within the HR system thereby identifying the individual experts and increasing the size and visibility of the overall pool of experts. Once registered as an expert, an individual can respond to queries and help others in areas of his/her expertise. The pool of experts can at a later stage review the content of knowledge within the system for its quality and relevance, and also identify and categorise certain knowledge assets as

premium assets for wider dissemination across the organisation. Through this system knowledge workers can declare, identify, and locate expertise within the organisation, and can use this information effectively to form project teams on the basis of the skills and competencies required to implement the project.

The intranet portal contains Project Profiles of all projects implemented by XYZ. These profiles contain details of the key contact person with regard to that particular project, and contact information of all individuals involved with the project, including clients. Therefore even though the senior management considers this system to be independent of people, it still is a system about people. In other words, it is a system that provides information about people that enables formal and informal contact across the organisation. The HR Management System for a particular project contains the team member identification, and the regular update for that member's skills, competencies and training profile. The system carries information that is not more than 10 years old, as older information/knowledge is not considered to be relevant to current projects. The project profiles are updated regularly especially in modules like QMS (Quality Management System).

## **8.5 Individual Roles**

Individual roles (C2-A16) are assigned by mapping the project requirements against the Competency and Skills Management System, which is an integral part of the knowledge portal. The system contains quantified competency levels, role suitability, role and individual profiles, re-evaluation reports, and the current and desirable competencies of individuals, and also sets individual targets for personal development. Routine activities are directed towards the development and adaptation of the operating processes of developing software. These activities are considered difficult for rivals to copy, and hold knowledge even after individuals leave the organisation. The activities attempt to organise how individuals work and interact. However individuals work differently, and have different preferences and tolerances for formal controls, Berkun (2005). Therefore within XYZ, in an attempt to make individuals more efficient and effective in their tasks, processes define roles and responsibilities that individuals are expected to execute.

Every function, position, and person at XYZ has clearly defined roles, responsibilities and objectives. Clear quantitative performance measures for each role are established. The performance measures are further classified into smaller, measurable parameters, which are



assigned differing parameters depending on the functions and level of individuals being assessed. These parameters include:

- leadership
  - proactive
  - takes initiative
  - followed by others
  - team player
  - takes ownership
- self development
  - quest for knowledge in technology
  - learns beyond work environment
  - attends training regularly
- technology initiatives
  - grasp of new products and technology
  - disseminates knowledge to colleagues
  - initiates new technology and systems
- quality management
  - reviews
  - benchmarks processes
  - benchmarks performance
  - improves and spreads quality objectives
- customer satisfaction
  - improvement in customer satisfaction index

The above qualitative parameters for performance measurement motivate individuals to share and contribute knowledge, learn continuously and develop their competencies and skills, while reinforcing the importance of customer satisfaction and quality.

Weekly review meetings at the project/supervisor level evaluate the reasons for non-achievement of individual task commitments. These meetings resolve issues at a cross-functional level where operational information necessary for the working of project managers and functional executives is shared. Monthly group/department level meetings review the roles and responsibilities. Quarterly staff meetings involve all employees and facilitate social interaction. The minutes of these meetings are recorded and shared promptly. Continuous feedback on project implementation can be obtained through these

regular, periodic meetings, internal employee satisfaction surveys, and external customer surveys.

## **8.6 Issues faced by XYZ as a Distributed Organisation**

XYZ is a large distributed organisation (C9) with offices located in more than 50 countries. The organisation therefore needs to integrate the issues and peculiarities of knowledge flow across different cultures and work practices. The issues facing XYZ in such a global distributed organisational front are adjusting to geographical dispersion of knowledge workers, loss of knowledge communicating richness (C9-A2) with less face-to-face interaction, coordination (C1-A6) breakdown in project management functions, loss of 'team ness' (C9-A4) as some teams are not co-located, and dealing with cultural differences among globally distributed staff (C9-A5) and units of the organisation. XYZ' main development centres were primarily based across India, while the organisation had sales and delivery centres in countries where the customers were located. However, in the recent past XYZ has initiated development centres also in the customer countries or low-cost countries in close geographical proximity to the countries in which the customers are located. This, initially led to a situation where the knowledge flow within these XYZ development centres was similar to a client-server model, that is, knowledge from all development centres would flow towards the base country, India, from where it would again flow out towards the dispersed country centres. For example, knowledge from the Santiago centre in Chile would flow to the main offices in India, and then if required, flow to a centre in Mexico City. Thus, XYZ faced the possibility of a loss of knowledge richness in transit. The presence of such issues were epitomised in quotes made during interviews conducted in February 2006:

*“When co-workers are not located in the same location, the important informal aspects of teamwork co-ordination such as socialising and camaraderie are harder to cultivate.”*

*“XYZ employees interact globally with clients or on-site development teams but no social ‘Communities of practice’ are practically possible on the global scale.”*

During further discussions about this issue the researcher recommended a possible employee rotation not limited only to project implementation but also temporarily co-locating team members, during project start-up and periodically through site visits and travel. Further suggestions included the concept of local knowledge champions (C9-A6) at

the branch level, and designated liaisons that are acculturated and informed about technical issues, as focal points of coordination and can transfer knowledge to other sites through regular travel visits.

During an interview conducted in February 2007, a senior XYZ management consultant confirmed that XYZ has followed the advice suggested by the researcher and now has therefore established knowledge champions at the local, branch and regional levels. Moreover, XYZ has attempted to improve the knowledge flow between regional centres by appointing local knowledge champions who meet regularly at the regional level, thereby increasing the coordination, communication, and frequency ((C9-A7)) of regular individual visits between such regional centres. Such visits and coordination between regional centres are also intended to acculturate individuals and make them aware of problems and solutions in development issues at the immediate regional level, and increase social interaction between the centres. The interview also confirmed that XYZ was attempting to supplement formal routines with collaborative social processes to promote effective dissemination of knowledge and organisational learning. With the intranet portal usage increasing 'exponentially', Communities of Practice and Centres of Excellence were now forming, at both the regional and corporate levels, and becoming an integral part of knowledge exchange within the organisation. Observations and interviews revealed that XYZ as an organisation is continuously learning from its environment through industry best-practices, cross industry best-practices, customers, competitors, visiting award winning organisations, from suppliers/vendors, institutes, reports collated within the KMS which depend upon the sensitivity of context, and white papers published in the KMS.

## **8.7 Measures**

XYZ needs to develop measures regarding the impact and benefits of implementing the knowledge management initiative. Currently the greatest indicator of the benefits of implementing the knowledge management initiative is Customer Satisfaction Index (CSI) (C12-A10). Senior management is of the view that the organisation is delivering better quality software quicker to their customers. They conduct weekly meetings to assess how the system is benefiting the organisation.

The senior management also expects a considerable benefit or return-on-investment for the knowledge initiative. As mentioned before, the knowledge initiative at XYZ evolved over the years therefore it would be difficult to reach an exact figure in terms of investment.

Also, there are no measures in place to analyse the benefit in terms of the systems implementation. The CKO will be developing parameters and criteria to measure the impact of implementing the KMS.

However, there is a change in the work methods of the organisation's business processes. The benefits that accrue are from the business perspective, that is, implementing the knowledge initiative has brought down the costs of developing software, and knowledge and artefacts are being reused without the need to spend additional time validating them. Therefore less effort is required in developing software and fewer artefacts are required to be developed overall offering further savings and increased reliability made possible through the effective sharing of knowledge.

## **8.8 Conclusion**

This chapter presented an evaluation of the findings of the case study conducted at a large software organisation. The findings provided insights into the organisation's knowledge vision, strategy, initiative development and processes. The chapter discussed the issues faced and addressed by the organisation during the development of their knowledge management initiative. The case study also provided insights into the issues of implementing a knowledge management initiative in a distributed environment. The chapter analysed the flow of knowledge within the organisation, and the infrastructure that supports and facilitates this flow. The empirical study also enabled the research to gain insights into the work practices and knowledge culture of the organisation. The insights can now be linked to the research proposition to shed further light on the flow of knowledge, as enabled and facilitated by the knowledge management initiative, during the software development activities of the project organisation.

## Chapter 9 Validating the Dynamic Feedback Model

The hypothesis for this research was formulated as an extension and adaptation of a published model, the DFM. The model was extended to incorporate the flow of knowledge, feedback and interaction. This chapter attempts to validate the DFM by identifying the functional areas and their interactions, and the flow of knowledge within the software projects of XYZ. The analysis presented in this chapter is based upon the empirical evidence provided by the case study. The previous chapters presented the findings and insights gained while conducting the case study. The chapters discussed the KM initiative at XYZ and how knowledge, including tacit knowledge, flows within a software organisation as a result of implementing this knowledge management initiative. Chapter 5 presented the DFM that provides a systems approach for facilitating and harnessing the flow of knowledge within software projects. Such leveraging of knowledge from one project to subsequent projects provides a continuous and long-term perspective to effectively implement projects. The model facilitates the flow of knowledge within the functional areas of project management, decision-making, technical development and quality assurance. This chapter identifies and analyses the flow of knowledge within and between these areas during the software development activities of the project organisation. The discussion in this chapter also emerges from the categories and attributes of Table 7, and their first appearance is marked in brackets.

### 9.1 Functional Areas and their Knowledge Flows

Interaction (C1) and feedback (C12) are major facilitators of knowledge creation, transfer, application, and learning (C12-A3) within a software project organisation. An important premise of this research is that the DFM model facilitates such interaction and feedback and provides the support required for knowledge to flow during the activities and practice of software development. The DFM identifies the four functional areas of project management, decision-making, technical development and quality assurance, and the feedback interactions between these areas, while the adapted version of the model further distinguishes the knowledge flows within and between the four functional areas.

This chapter identifies the functional areas of software development within the organisation in which the case study was conducted. The flow of knowledge within and

between these functional areas is analysed, and the different modes and mechanisms of interaction and feedback are further analysed to identify the support they provide. The organisation's knowledge flow is then compared with the adapted version of the DFM to confirm and validate the model.

### 9.1.1 Technical Development Area

XYZ is continuously implementing software projects. Being an ongoing activity (C11-A1), knowledge (C3) also continuously flows between these projects, and the functional areas of each project. Knowledge is created within the technical development area while executing the activities of eliciting and finalising requirements, writing specifications, designing the software architecture, coding, and reviews (C3-A2). The activities of technical development also provide individual members of the project teams with experience which they are able to use for decision-making and problem solving while implementing subsequent projects. An interesting observation was that XYZ is a CMMI level 5 organisation, which implies that its software development processes are standardised and have formal routines that optimise and discipline the process and aim to be not "*dependent upon people*".

However an important observation elicited during a discussion with experienced developers was that they preferred to make less use of formal routines, control and discipline (C7-A1). An unofficial query, that is, not during a formal interview, regarding the observation was answered along the lines that formal routines tend to be too rigid and prescriptive, and thus stifle creativity, flexibility and informal learning that are an important aspect of software development. The discussion progressed towards the control and discipline versus creativity and flexibility (C11-A14) debate regarding software development processes. The view was that an emphasis on control and discipline leads to "*project managers managing the process rather than the project,*" and "*when processes are required to manage processes, it's hard to know where the actual work is being done.*" Therefore, instead of focusing on processes and methods, project managers need to focus on their teams and "*make each individual aware of their role in the process.*" Simple planning or tracking systems should be used but they must match the complexity of the project and the culture (C10) of the team, that is, planning and tracking should support the team in achieving project goals and not inhibit them. A developer stated that "*actual software practice is far different from theoretical academic recommendations*", and as an

example recounted that most developers would rather ask a colleague than refer to a coding manual or process guide when faced with a problem (C10-A6).

The above example, while highlighting social interaction (C1), exchange of tacit and explicit knowledge (C3-A8), and learning from peers (C3-A10), also expressed the degree of flexibility, adaptability and freedom software developers prefer while executing their tasks. The developers were of the view that software processes within the technical area should be designed and structured to enhance creativity, flexibility, adaptability and initiative rather than inhibit them. Process discipline (C6-A12) is also required because team members have diverse levels of skills, expertise and knowledge, and also a wide variation in “*process maturity*”, and therefore the common purpose of the process could be open to “*wide*” personal interpretation amongst them. The knowledge level of individuals enables them to make decisions during software development, while process discipline brings in the element of “*standardised control*” required to ensure that there is not too much variance in the process output. Also, knowledge flow (C3) evolves during the process and flexibility is required to facilitate this flow to ensure that it is available to individuals who require it, especially when the life of this knowledge might be very short. The discussion revealed that team members did document the processes as learnt in their training (C2-A1), and most processes were followed as a normal way of performing tasks, that is, the processes were internalised and institutionalised within the development activities of the technical area. The essence of the views expressed can be highlighted with a quote from the discussion which states that “*disciplined software processes should enable the development team to ensure the quality of the product, and while encouraging creativity, should not be restrictive and prohibitive.*”

### **9.1.2 Quality Area**

The quality (C8) area is where mismatches in the software product’s functionality are identified and defects (C8-A1) uncovered, and errors committed in the technical area are corrected. This necessitates a review of the requirements, code and test cases leading to an analysis of the causes of defects and mismatches. In XYZ, the causal analysis (C8-A6) report is discussed in defect prevention meetings that review and update the checklists (C8-A9) and templates to ensure that coding standards are maintained. New learning (C3-A20) emerges in the quality area by addressing the defects and problems that are discovered, and as suggested by one project manager’s interview quote “*quality patterns are a common way of learning and capturing experience.*” Best practices (C6-A11) and lessons learned

(C3-A20) are identified within the area, stored in the Quality Management System (QMS) (C8-A4) asset library, and published and disseminated throughout the organisation by the portal (C8-A10). The causal analysis report is also logged into the Integrated Project Management System (C8-A6) so that the project management group and the SEPG are aware of the defects and context of issues such as requirements and design, in case of a similar occurrence in another project, or during the start-up of a similar project. The quality area is an excellent area for training new recruits as it is considered important that they are made aware of the mistakes, defects and causal analysis at the beginning of their software experience and would therefore be expected not to repeat the mistakes they initially helped in uncovering (C8-A14). Quotes from interviews that emphasise the same include:

*“Employ freshers in QA (Quality Assurance) before coding. It will be very beneficial as they do the causal analysis and understand what went wrong.”*

*“Future employees need to face causal analysis during their training.”*

The outcome of providing insights into quality improvement during training would be reflected while testing and result in increased reliability and performance of the software product.

### **9.1.3 Project Management Area**

As the quality reviews are conducted, the checklists and templates (C8-A8) are also reviewed, updated and reported within the IPMS (C5-A4) and associated decision-making analysis tools (C4-A4). Thus, learning from the quality area flows into the project management and decision making area. Project planning (C4-A5), monitoring (C5-A13), control (C5-A13), closure (C5-A11), team selection (C2-A17) and training (C2-A1) take place within this area. Knowledge is required and exchanged while preparing contracts, proposals, scope statements, budgets, workable requirements, work breakdown structures, estimates, and schedules. During interviews, when asked if project managers performed any knowledge activity while executing activities during the initial stages of a project, most replied that they did not. However, further indirect questions and observation provided insights into their functioning during this phase of the projects and revealed that for almost all tasks project managers were engaged in a knowledge activity or referring to a knowledge source. Project managers almost immediately referred to the portal for project start-up activities, and either interacted with a colleague who they were aware had



implemented a project with similar attributes and characteristics, or located and contacted an expert, again either through the KM-P within the portal or upon a referral. Thus an important observation emerged that while knowledge workers, in this case project managers, consider knowledge-management is a priority, they assume that they do not seek knowledge management support at the beginning of a project. However, observations and inferences from interview data reveal that in actual practice project managers perform knowledge activities at the start of a project.

An important knowledge activity performed in the project management area is ensuring that the right skills, competence and knowledge are available to implement the project. The personnel requirements of experience, skills and technological knowledge are submitted to the Manpower Allocation Task Committee (MATC) (C2-A8) which further refers to the Human Resources System (HRS) (C2) to check the availability of the required competencies (C2-A2). The HRS is able to ensure that individuals with the appropriate skills and knowledge are short-listed for the team selection process. Competencies are updated on a weekly or monthly basis within the HRS depending upon the training provided which is always planned for roles. Based upon proficiency levels, personal development programmes are tailored for each individual and the capability changes are reviewed and monitored. The training (C5-A12) ensures that explicit knowledge from the organisation flows to the individual who internalises it and then applies it to execute tasks while implementing projects. The primary objective of the HRS is to identify individuals for different and specific roles for existing, new and upcoming projects. Team selection for specific expertise is determined with the help of a tool called 'Team Track' (C2-A9). Since late 2006, this tool is now being used more frequently as XYZ improves upon its vertical silo structures and lays more emphasis on cross-functional and, even more recently, virtual teams. The designers of Team Track are attempting to introduce a feature to ensure that tacit knowledge, possessed collectively by the team members flows along with the individual person assigned to each specific task or responsibility, that is, the individual's role in the process. In essence, the tool will attempt to identify which individual in the development process is best qualified to apply his or her knowledge to a specific process task.

A popular recommendation within project management literature to ensure individuals' commitment to the project team and implementation is to involve team members during the project planning process (Lewis 2002, Richman 2006, Dalcher and Brodie 2007). Project planning is generally accomplished by small groups responsible for managing the

project and results in creating new knowledge, whereas project implementation knowledge is principally associated with the individual executing the project implementation tasks. Thus, the knowledge flow associated with a team's planning capability involves ensuring that before the team can create and evaluate a project plan, its members must learn to work together effectively, develop a common set of terms, and establish the processes that will be used to develop the plans. XYZ has attempted to address these capabilities during the training provided to their employees where an emphasis on team building and commitment complements the technical training (C2-A1) provided. Observations showed that project managers begin project planning through the process of leadership (C2-A7), which drives tacit knowledge. This leadership knowledge is used to formulate an approach for communicating with other team members and enlisting their active participation in the planning process. Knowledge about what is expected of the other members is provided during the above mentioned training sessions provided to inculcate the individuals in XYZ' work methods and culture. The training enables the team members to comprehend and apply the project manager's suggestions for how the team is to perform and implement the project. Working together, the team then applies its collective tacit knowledge to synthesise plans and courses of action, which then become the focus of continuous (C11-A5) evaluation and refinement while implementing the project. Thus, XYZ has its own characteristic pattern of knowledge flow enabled by the processes of leadership and evaluation.

Mentoring (C2-A6) plays an important role in the flow of tacit knowledge at XYZ during project monitoring and control. During meetings it was common for project managers to review the decisions and actions of individual developers, and provide guidance for learning important techniques and improving their performance. This represented a flow of tacit knowledge at the individual level which takes place between the project manager, as a mentor, and the individual. New knowledge acquired by the individual is created through the project manager's evaluation of the individual's performance and actions, and feedback. The individual comprehends and organises the knowledge associated with the project manager's review and inputs. These discussions suggest a cycle of formalisation, synthesis, application, refinement, and evolution of knowledge and this knowledge becomes increasingly tacit as it flows. As this knowledge becomes internalised within an individual's knowledge base, the developer's describe it as combination of their individual knowledge and the project manager's experience. Thus knowledge flows from different processes such as experience and mentorship interact but only when the project manager

evaluates an individual's performance and provides a review and evaluation of the tasks performed.

Apart from the ongoing review and evaluation during project monitoring and control (C5-A13), key organisation learning for future improvements takes place during project closure (C5-A11). XYZ attempts to capture lessons learned to improve the software development process. However, as previously mentioned, the observations and interviews revealed that while best practices has a positive connotation, lessons learnt seemed to have a negative connotation within the organisation's culture. Therefore XYZ needs to perform this exercise carefully and in an atmosphere of safety for the individuals involved. The exercise is called After Action Reviews (AAR) (C5-A10) and seeks to answer four main questions:

- a. What was supposed to happen?
- b. What actually happened?
- c. Why were there differences?
- d. What can you learn from this experience?

Another important question that team members need to retrospectively consider is what tasks could have been executed in a better and more efficient and effective manner. A senior project manager was referring to such issues when he stated that the AAR's aim is *"to determine what went right, what went wrong, what worked, what did not, and how it could be made better next time."*

At XYZ the project manager carries out the closure analysis with help from members of the SEPG. A template for the analysis report is available in the portal and when duly completed with mostly metric data to ensure a focus on objective information, is stored in the central repository, thus becoming available to others within the organisation. The template contains parameters and information about the project plan, deviations, defects, and actual performance data for analysis. The template report is analysed by the quality advisor, who develops an initial interpretation of the information. This interpretation is discussed by the quality advisor with the project manager and other team members, and along with further inputs and suggestions a final closure analysis report is prepared. The final report is reviewed by the SEPG and is also available in the central repository. The SEPG is considered the forum for disseminating the learning from project closure reports. XYZ believes that the SEPG refines existing knowledge with learning from project closure analysis that leads to the creation of new knowledge. This new knowledge is organised by the SEPG to synthesise new process assumptions and then formalised as guidelines, that is,

the knowledge is made explicit and implemented within the organisation's software processes, thereby providing a knowledge flow that contains an evaluation of current and previous projects.

It was observed that the AARs and closure analysis (C5-A11) at XYZ address deviations to expected performance and activities. It was also observed that XYZ did not conduct project retrospectives which enable an organisation to address issues such as: what were the important informal relationships that existed during the project; how strong were those relationships; did they aid or hinder project progress; who does not get along with whom on the project team; what was the formal reporting structure; did it change over time and if so, how; who filled each position, and what were their responsibilities; when did they start in this role and when did they stop; and were the right skills and competencies available to implement the project.

Kerth (2001) states that 'reflecting upon a completed project is not a natural activity, but must be formalised.' However, a project retrospective (C5-A12) should not focus only on mistakes and participating individuals should feel 'safe'. Safety means that participants must feel secure within their project team to discuss their work, to admit that there may have been better ways to perform the work, and to learn from the retrospective exercise itself. An important aspect noted during the interviews conducted and observations made was that within the organisation's work culture, best practices have a positive connotation while lessons learnt are considered to have a negative connotation and are associated with post mortems, failed projects and project reviews. Therefore it is important that AARs, closure analysis or project retrospectives conducted by the organisation do not become complaint or fault-finding sessions, otherwise important learning will be missed. A project manager stated that "*I always ask my team members to work the problem and try not to ascertain blame or responsibility for it.*"

The aim of retrospective exercises is to help participants create an environment in which they articulate what is most important to them while reviewing the project for significant learning to determine what long-term activities need to occur for the next project to be more successful. Project retrospectives should include the entire project team, as no one person can know the whole story of the project, and an individual's reflection can only provide a small perspective on the learning. While individuals may learn from specific experiences, they may not be able to grasp other issues that require a larger perspective and the ability to understand the relationship between an individual's work and that of an entire

team. Kerth (2001) recommends that retrospectives help build trust within a project team, and help team members to find improved ways of working together. Kerth (2001) also states that 'project managers should participate, contribute, and learn rather than dominate the exercise, and help the team to understand what the real issues are in a humane, caring, and effective manner.'

#### **9.1.4 Decision-Making Area**

Project managers at XYZ have different views, perspectives and preferences regarding projects and their implementation. While deciding upon project methodology, design, architecture, tools, and technologies they refer to the updated checklists and templates within the IPMS and QMS. This depicts the flow of knowledge from the project management and quality assurance areas to the decision-making area where it is translated into action while implementing subsequent projects. Decision-making (C4) takes place while planning and executing software processes, and analysing the risk (C4-A8) involved, and is an area that drives the development activities. Decision-making is done by all members of the project team, SEPG, Quality Assurance Group (QAG), and other stakeholders. Knowledge is constantly shared through regular team meetings, social interaction, communication infrastructure and portal to make decision-making more effective. The tools used for decision-making include Grid analysis, Pareto analysis, Fishbone, SWOT analysis, Modelling, Simulation, Process Assets Library, 6 Thinking Hats, Central Repository, Visual Source Safe, Version Control Cases, Templates, Function Point analysis, Brain Storming, Fishbone analysis, Use Case Scenario mapping, Rational Rose, and Master Craft which has been indigenously developed by XYZ. These tools are used for decision-making during requirements analysis (C4-A6), project planning, budgeting, estimation (C4-A5), risk analysis (C4-A9), design (C4-A6), testing (C8-A11), and implementation. For example, the grid analysis method is used to translate Entity Relationship Diagrams (ERD) into data-bases thereby addressing future requirements and supporting subsequent extensions to the software, and brain-storming (C3-A11) is used during the testing phase during unit and integration testing and demonstrations. Brain-storming, grid analysis and fishbone method are used to decide upon whether to implement a change request at a particular stage or not. Project managers mentioned that they do have preferences with regard to which tool to use based upon previous usage experience and the tool's applicability. Knowledge created and applied within the decision-making area forms the basis of knowledge used within the technical area for developing software. This

knowledge flows through the software processes and permeates the activities of problem solving and technical development.

### **9.1.5 Summary of Functional Areas**

Based upon the concepts and theoretical propositions discussed in Chapter 5, this research identified the functional areas of the software development effort at XYZ. The functional areas identified are technical development, quality, project management, and decision-making. Observations and interviews conducted at XYZ provided data that when analysed seemed to depict the possibility of a fifth functional area of human resource management. However, further analysis established that human resource management was an integral function of project management, Kerzner (2004). Therefore the emphasis and importance of human resource management was included and discussed within the activities of the project management area.

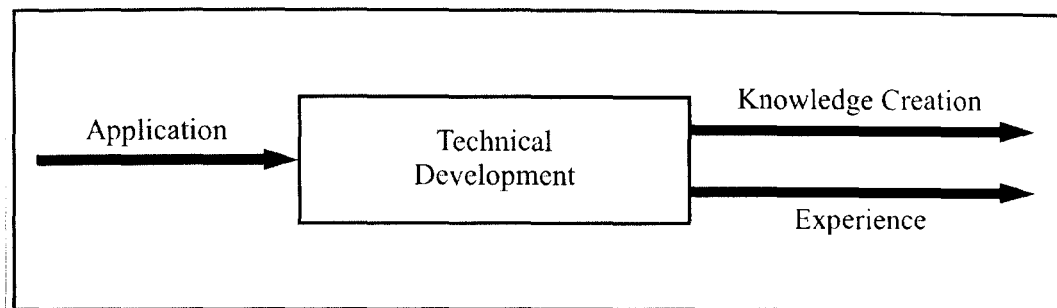
The discussion in this section identified the knowledge management activities within the various functional areas of the software development effort at XYZ. These activities included the sharing and exchange of tacit and explicit knowledge. The following sections analyse the interactions and interdependencies of these functional areas and the relationships between their knowledge flows.

## **9.2 Relationship between Knowledge Flows and Functional Areas**

The previous section identified the functional areas and the knowledge flows within them as observed in the software organisation where the case study was conducted. The interactions between these knowledge flows and functional areas depict the overall complexity of the development effort, and integrate the activities and processes of developing software, project management and facilitating the flow of knowledge. The flow of knowledge between the functional areas can be depicted in diagrams that model the relationships as dynamic systems of nodes and arrows. Nodes represent the functional areas, while arrows represent the relationships between these functional areas.

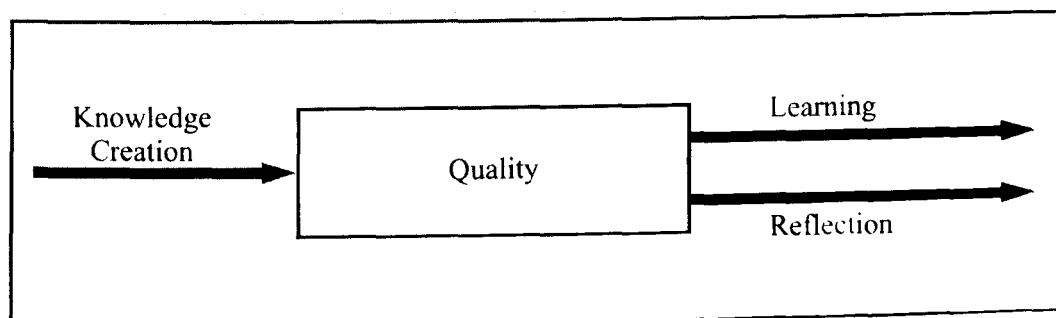
Therefore, as mentioned in Section 9.1.1, XYZ is continuously implementing software projects on an ongoing basis. The existing knowledge of team members is applied, and further knowledge is created during problem-solving and engaging in activities of technical

development while developing the artefact within the technical area. This process of knowledge creation and integration, creates project and product specific knowledge, and also enhances the individual and collective team experience (C7). A developer's response in a questionnaire was "*knowledge about the product and domain is acquired during the technical development of software,*" while a project manager's response was that "*experience helps in understanding problems and creating effective solutions.*" Thus the output flow from the technical area is the creation of new product knowledge and an enhancement of individual and team experience. A diagrammatic representation of this relationship is provided below in Figure 7.



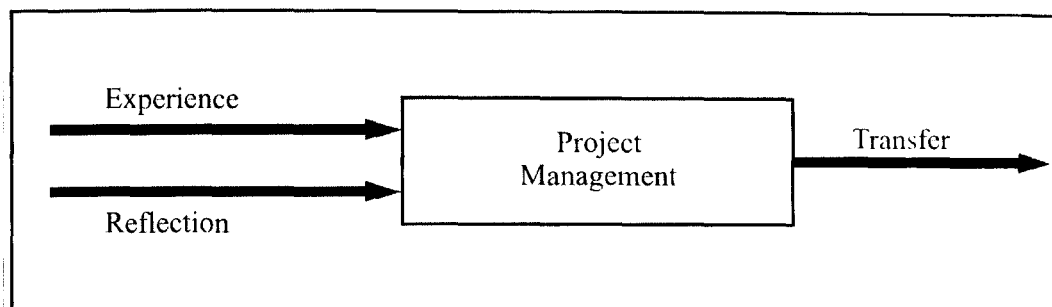
**Figure 7 Inflow and Outflows for Technical Area**

The project and product specific knowledge flows to the quality area where it is applied to identify mismatches and detect defects in the product. New learning emerges in this area when errors are corrected, and knowledge is also created while analysing the defects to ensure that the product conforms to the required specifications. The view is supported by a project manager's response that "*impact of the modification done after addressing mismatches is accessed on the whole system and the changes are made throughout the system to minimise further mismatch issues on implementation*" The new knowledge created further integrates with the existing knowledge when updating checklists and performing causal analysis. Thus the quality area benefits from the product specific knowledge created in the technical area and provides further learning and reflection (C8-A3). These flows are presented in Figure 8.



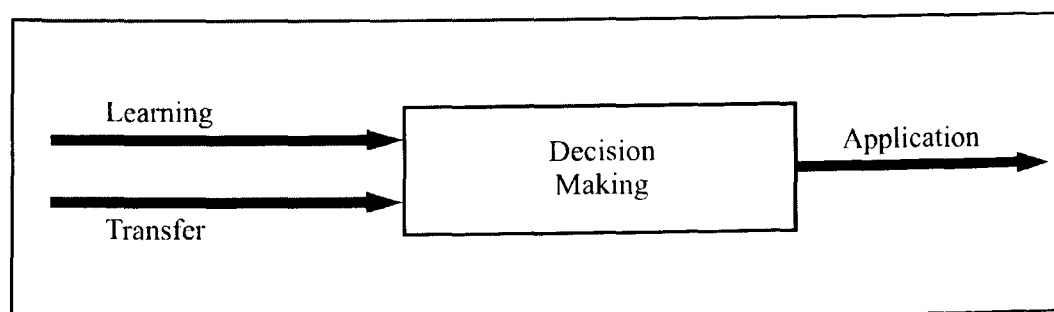
**Figure 8 Inflow and Outflows for Quality Area**

The project management area benefits from enhanced experience gained in the technical area, and from the further reflection provided by the quality area. The project management area integrates such experience and reflection by updating project management templates and modules to ensure more effective planning, control and monitoring of projects. Integrating experience and reflection creates further dynamic knowledge that the project management area is able to transfer to the decision-making area. Responses received during interviews and in the questionnaire state that “*reviews*” and “*experience gained*” while implementing a project help “*improve project management processes*” for subsequent projects. Figure 9 presents these relationships, highlighting the project management area’s focus on the transfer of knowledge.



**Figure 9 Inflows and Outflow for Project Management Area**

The decision making area benefits from product specific learning from the quality area and the dynamic knowledge from the project management area. This enables more effective decision making that is applied within the technical area for current and subsequent projects. For example, as questionnaire respondents state that “*knowledge acquired*” while implementing a project enables “*better planning*” and “*better software designs*” in future projects. The decision-making area is concerned with the application of decisions, and this is represented in Figure 10.



**Figure 10 Inflows and Outflow for Decision-Making Area**



### 9.2.1 Interactions of the Relationships between Knowledge Flows and Functional Areas

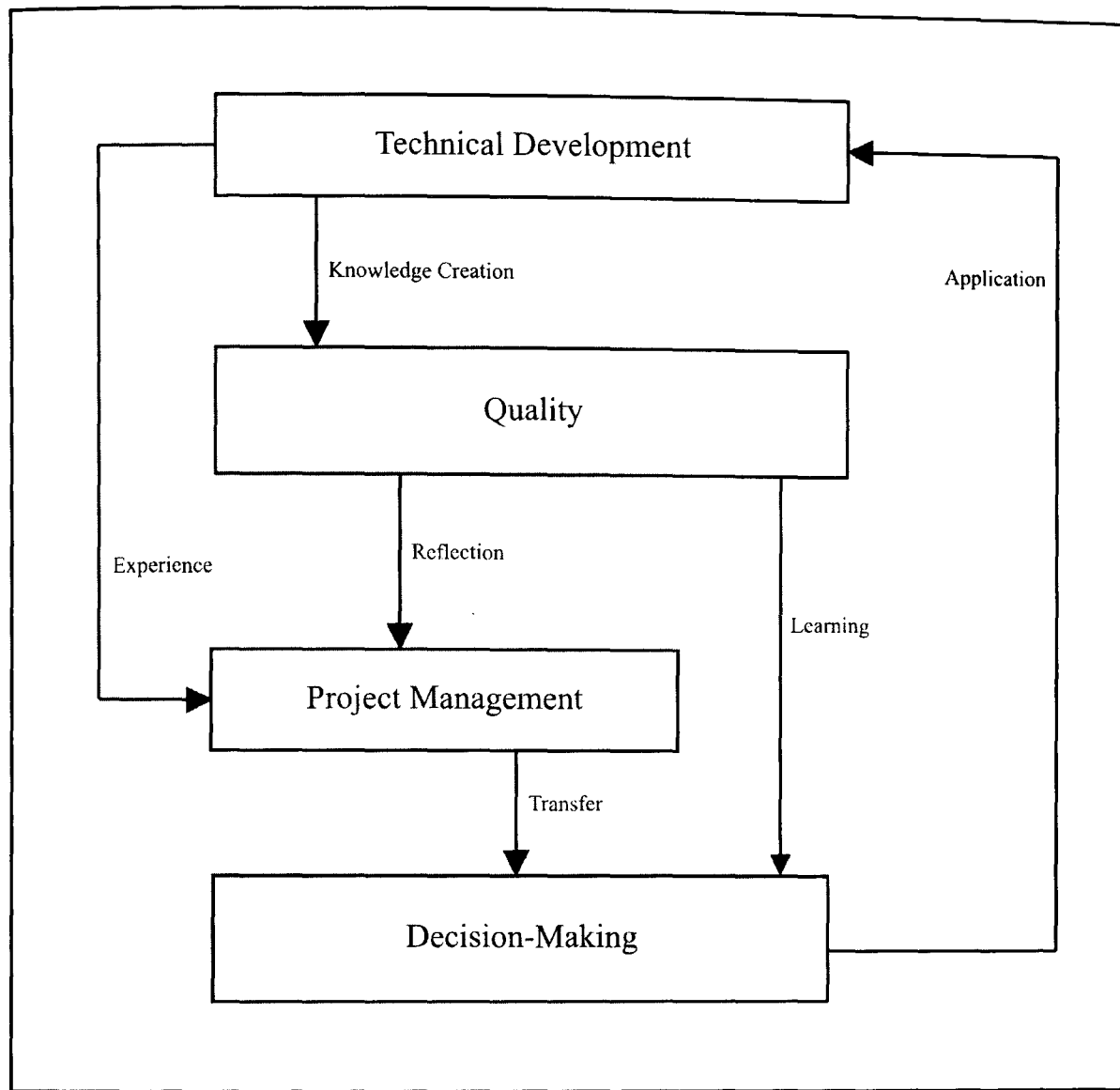
Table 9 below provides a summary of the relationships between the functional areas and the knowledge flows as analysed in the previous section along with the actions that link them. However, it is important to note that the activities attributed to each functional area are not exclusive to that area, but depict a relationship where the emphasis on that activity is greater than other activities, within that particular area.

**Table 9 Relationship between Knowledge Flows and Functional Areas**

<b>Actions</b>	<b>Technical Development</b>	<b>Quality</b>	<b>Project Management</b>	<b>Decision Making</b>
Knowledge Creation	*	*		
Learning		*		*
Reflection		*	*	
Knowledge Transfer			*	*
Knowledge Application	*			*
Experience	*		*	

The activities of knowledge creation, learning and reflection clearly play an important role in uncovering and reviewing mistakes and mismatches. Knowledge created while developing the software product is applied to identify mismatches, while performing causal analysis and ensuring that such mistakes are not repeated, enhances learning and reflection. Such reflection is required along with the experience of previous technical development while planning subsequent projects within the project management area, and knowledge created in doing so, needs to be transferred to the decision making area to ensure that it can be applied during future problem solving. Knowledge and learning gained through problem solving during decision making need to be applied while developing software in subsequent projects, while the knowledge created and experience gained need to be fed to the quality and project management areas for further application. Thus a continuous and iterative (C11-A4) flow of dynamic knowledge is facilitated during the software development activities of the project organisation. This flow of knowledge within and

between the functional areas while developing software at XYZ as analysed above is represented below in Figure 11.



**Figure 11 Knowledge Flow within and between Functional Areas**

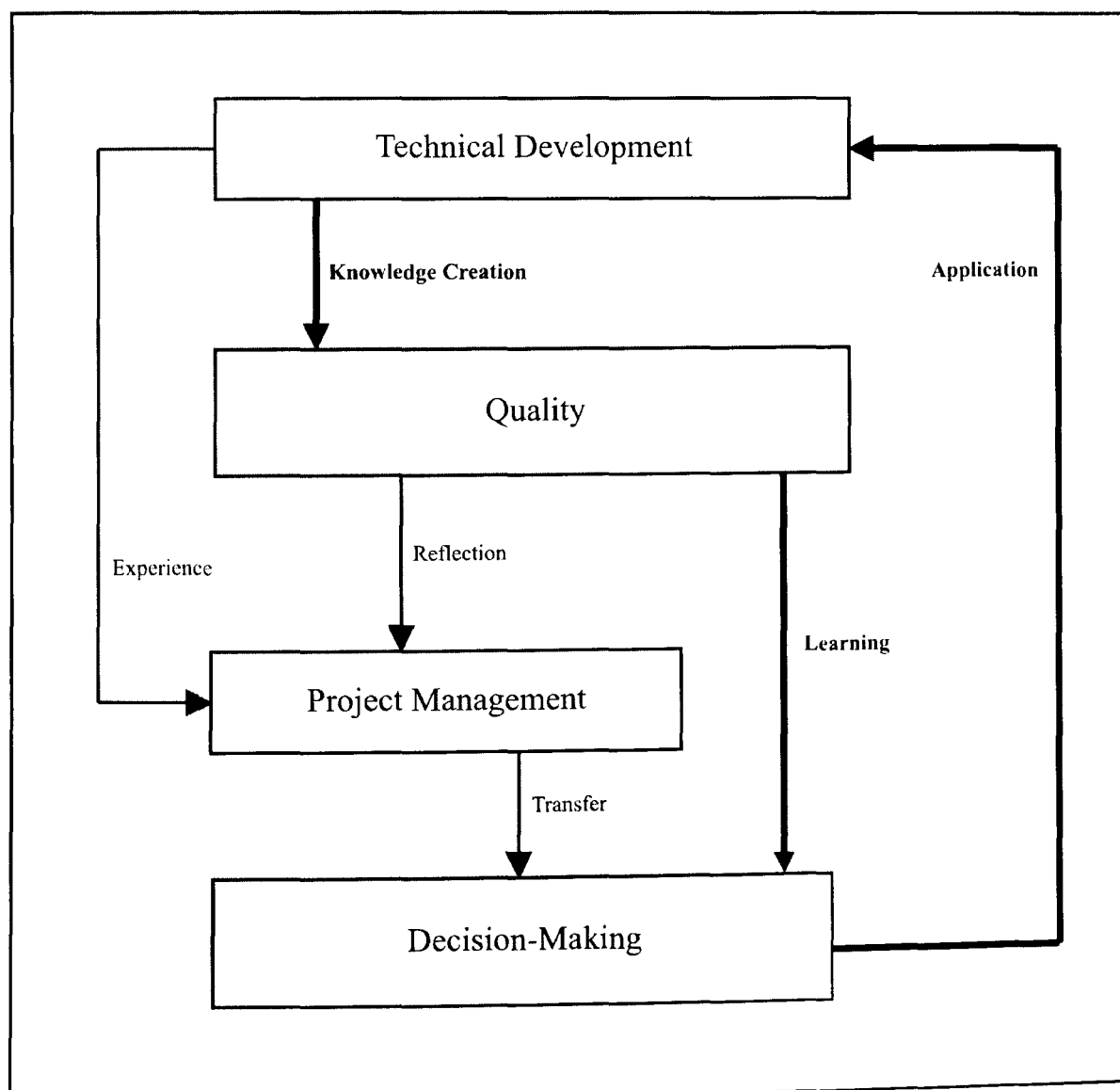
### **9.3 Feedback Loops**

The relationships depicted in Figure 11 are in essence a set of interactions and feedback loops depicting software development activities from a continuous and dynamic perspective. The feedback cycles complete the dynamic loops that enable the functional areas to operate in a dynamic manner, and can be analysed in terms of causal-loop diagrams.

#### **9.3.1 Knowledge Creation--Learning--Application Loop**

The *knowledge creation-learning-application* loop enables new product specific knowledge that is created, to be used to detect and correct defects and errors, and the

application of the learning that emerges while analysing the defects. This loop is therefore concerned with new knowledge creation (Nonaka and Takeuchi 1995), learning (Kolb 1984, Piaget 1970) and the application (Vera and Crossan 2003) of this new learning and knowledge. Therefore, this loop connects the nodes or functional areas of technical development, quality and decision-making. The loop enables the flow of knowledge created within the technical area to the quality area. The case study evidence at XYZ established that identifying mismatches and performing causal analysis result in new learning, and the loop facilitates the flow of this learning to the decision making area. Learning from the quality area helps in subsequent decision making to avoid repeating mistakes, and helps in preventive action by identifying and prioritising risks, Pressman (1997) and Sommerville (2003). The loop enables knowledge generated in the decision making area to be applied in the technical area while developing subsequent software products. This loop and the flows it facilitates are highlighted in bold typeface in Figure 12.



**Figure 12 Knowledge Creation—Learning—Application**

### 9.3.2 Knowledge Creation--Reflection--Transfer--Application Loop

The *knowledge creation-reflection-transfer-application* loop facilitates interaction and flow of knowledge between all four functional areas. The loop enables product and project specific knowledge to flow from the technical development area to the quality assurance area. Causal analysis of errors and defects detected and corrected in the quality area provides new insights and learning, which combined with the product knowledge further provides integrated knowledge that needs to be deliberated, reflected and acted upon. The knowledge creation-reflection-transfer-application loop ensures that such reflection (Kolb 1984, Dyba 2003) and translation is done in the project management area, where existing templates and modules are modified. The consequent effects of such modifications are transferred to the decision making area where they are effectively implemented, and further applied in the technical area, through flows facilitated by the loop. These flows are highlighted in bold typeface in Figure 13.

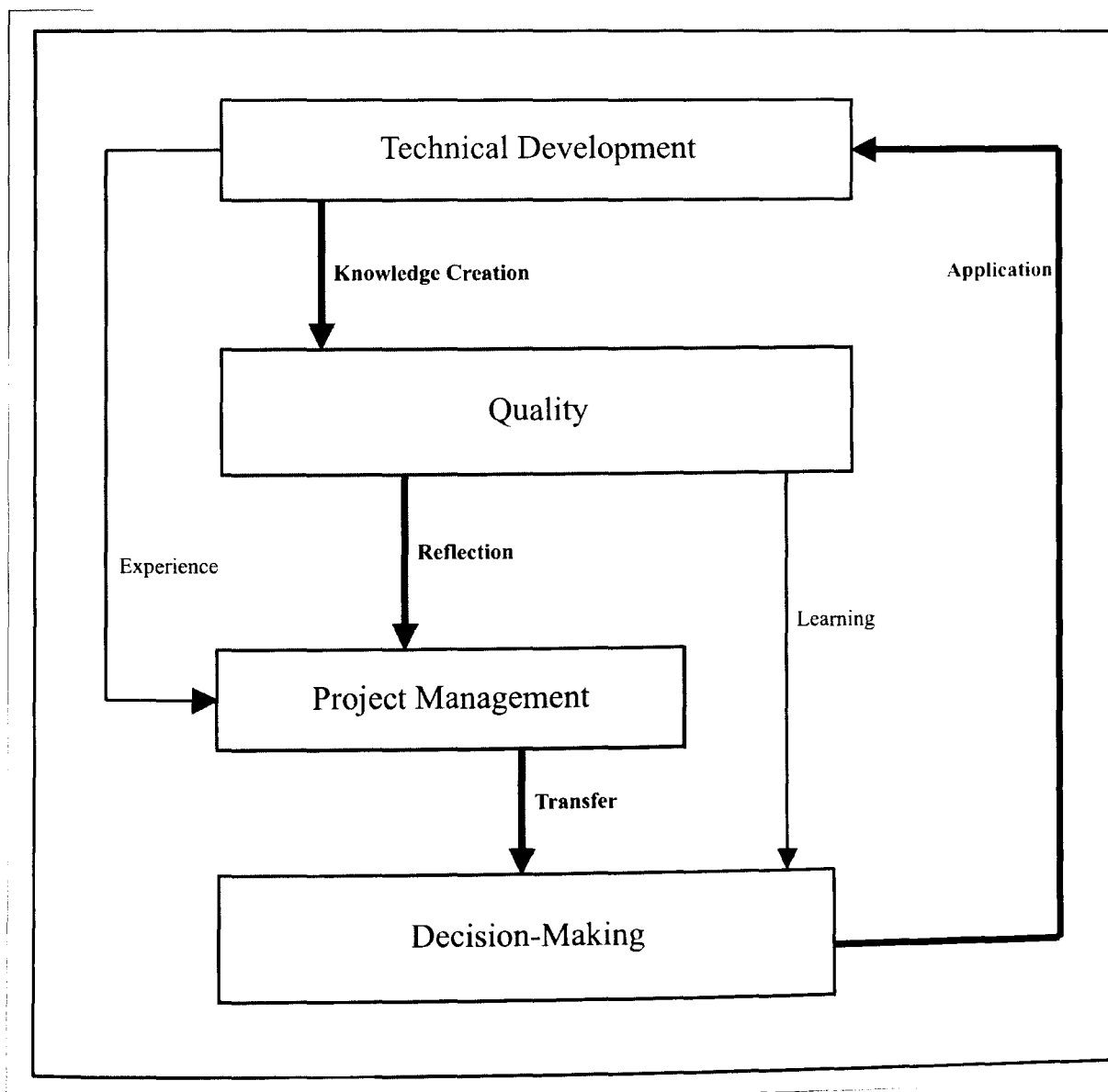


Figure 13 Knowledge Creation—Reflection—Transfer—Application Loop

### 9.3.3 The Experience--Transfer--Application Loop

The *experience-transfer-application* loop facilitates interaction and the flow of knowledge between the technical development, project management and decision-making areas. New knowledge is created while developing the software product and this knowledge integrates with the existing knowledge of individuals and team members (see Section 9.1.1). Such integration of knowledge enhances individual and collective team members' experience. Grant (1996). The loop facilitates the flow of this experience (Kolb 1984, Piaget 1970, Dyba 2003) to the project management area where existing assumptions and plans are modified. The insights gained by such modifications are transferred to the decision making area for effective implementation, and flow to the technical area to be subsequently applied in the development process. **This loop depicts the dynamic nature of the flow of knowledge, and how experience gained while implementing a project can be effectively transferred and applied to subsequent projects.** Thus the flows represented by this loop effectively capture the essence of the hypothesis and research question as stated in Chapter 5, as presented below in Figure 14.

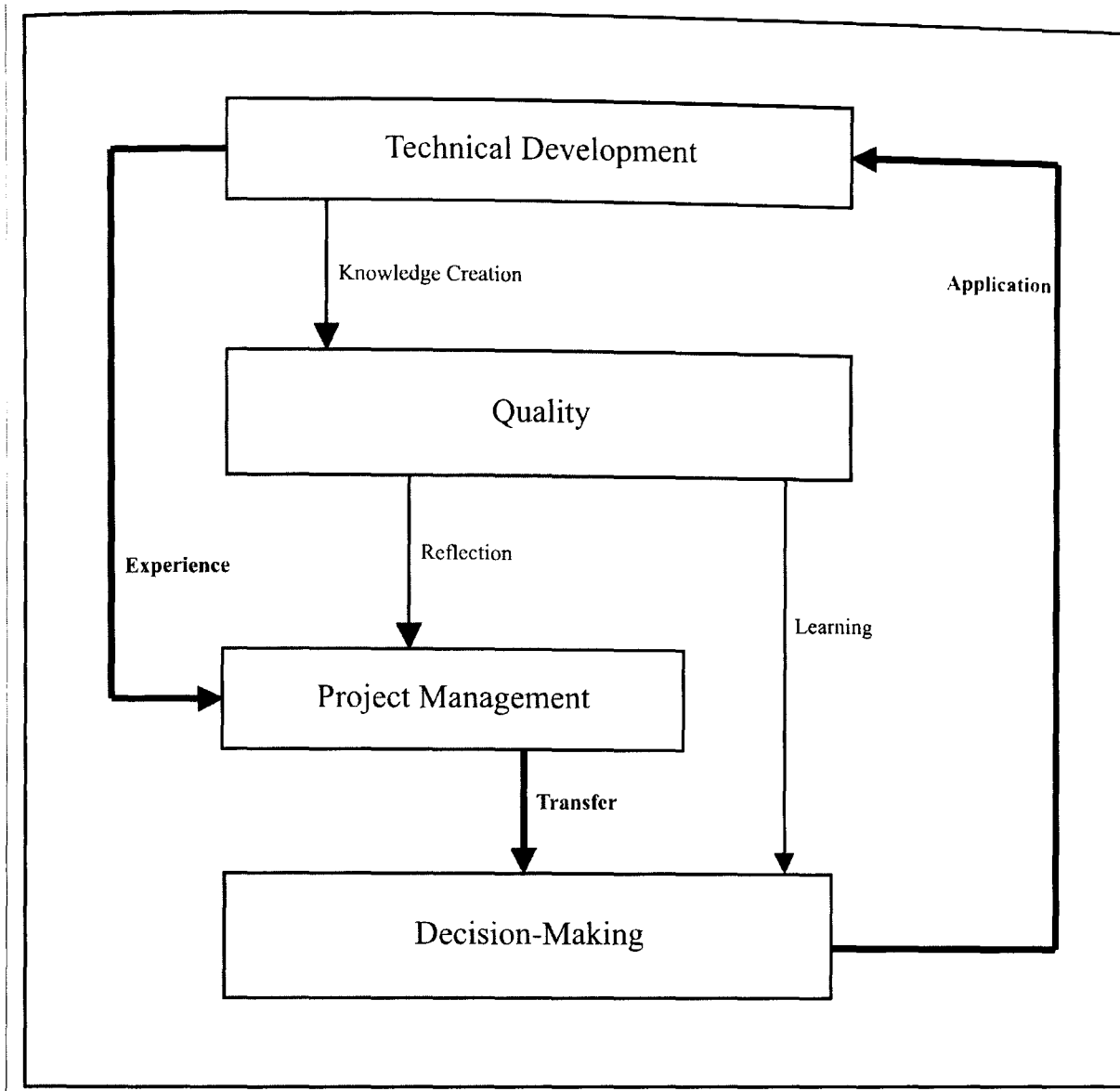


Figure 14 Experience—Transfer—Loop

### 9.3.4 Summary of Feedback Loops

Based upon the relationships between knowledge flows and functional areas, as discussed in Section 9.2, this section analyses and discusses how the dynamic flow of knowledge is enabled by feedback loops. The feedback loops complete the set of interactions and interdependencies between the various functional areas. The feedback loops enable new learning and knowledge, both tacit and explicit, to be available for use when required in the software development process. The loops also ensure that new knowledge integrates with existing knowledge in a dynamic manner, while mistakes and errors are reflected upon to improve the development process. Finally, the feedback loops of the model ensure that experience gained while implementing a project is effectively transferred and applied to subsequent projects. Thus the model facilitates the dynamic flow of knowledge within the functional areas of the software development effort.

## 9.4 Knowledge Flows, Functional Areas, DFM and the K-DFM

The knowledge flows and functional areas as analysed in the previous sections appear similar to those proposed by the DFM. The evidence of knowledge flow at XYZ shows that it is possible to identify and categorise the knowledge activities of software organisations within the functional areas of project management, decision-making, quality, and technical development. Software project organisations design their operations and processes around functional areas, and it is the flow of knowledge between these areas that continuously helps improve the effectiveness and success of the organisation's development activities. As mentioned in the previous section, it is important to note that the activities and processes that facilitate and support knowledge flows are not exclusive to these areas but are more dominant and a greater emphasis is placed on them within these areas. Also, while it is possible that other functional areas presented by alternative perspectives could also possibly be identified, the research was able to identify the functional areas as proposed by the DFM, and the flow of knowledge within and between them. Thus, the research validates the theoretical propositions of the DFM by identifying that these functional areas and the knowledge flow between them exists within the daily operations, practices and processes of developing software in a large project organisation such as XYZ. The research specifically validates the proposition that experience from executing a project can be transferred and applied for more effective implementation of subsequent projects.

The feedback loops that facilitate the transfer of learning and knowledge as depicted by the Knowledge Dynamic Feedback Model (K-DFM) in Figure 11 provide double-loop feedback, (Argyris 1976). Double-loop feedback relies on periodic reassessments of standards and norms to ensure they remain relevant, thus enabling the evaluation of organisational assumptions in order to improve capabilities. The dynamic and iterative nature of the knowledge that flows within the feedback loops of the K-DFM facilitates the reassessments of the processes and governing variables within the functional areas to ensure their relevance and effectiveness. The following section reviews the knowledge flows of the K-DFM as observed in XYZ and evaluates them with Grady's (1997) PDCA process improvement cycle, as discussed in Section 5.3.

#### 9.4.1 Evaluating K-DFM with Grady's PDCA Software Process Improvement Cycle

Process improvement helps make software processes more efficient, effective and productive. Reported benefits of continuous process improvements within software projects include: a 35% increase in productivity; decreased costs and increased ability to meet budgets; increased on-time deliveries; increased product quality as depicted by a 39% reduction in error rate; increased customer satisfaction; and, increased staff morale. (Dalcher, 2005c). The following discussion evaluates the K-DFM with Grady's (1997) PDCA cycle of software process improvement to determine the model's functionality and potential to improve processes.

Section 9.2 discusses the relationships between knowledge flows and functional areas in XYZ. Significant quotes from the data collected at XYZ established that "*knowledge acquired*" while implementing a project enables "*better planning*" and "*better software designs*" in future projects. These quotes helped determine that as knowledge gained from one project is reflected upon and enhances an organisation's collective experience, it helps improve planning for subsequent projects. This is depicted in the relationship where knowledge is transferred from the project management to the decision-making functional areas of the K-DFM to make correct or improved choices about development plans. This knowledge is applied for better planning in the decision-making area similar to the 'planning' step of Grady's (1997) PDCA cycle, where choices are made about the objectives and processes necessary to deliver results in accordance with the requirements and specifications.

The improved planning provides new insights about the project to be implemented and the software product to be developed. These insights combined with the existing knowledge of the team members are applied within the technical area to develop the product, which matches the 'do' step of the PDCA cycle and is about implementing the process. Product development in the technical area creates new knowledge about the product which is utilised in the quality area for purposes of product assurance. The new knowledge also enhances the collective experience that further improves project management practices of the organisation.

The new knowledge that flows to the quality area helps determine the "*usability and functional defects*" of the software being produced. Correcting errors and ensuring that the



product conforms to the required specifications enables new learning and further knowledge creation, and is similar to the ‘check’ step of the PDCA cycle where processes are monitored and results evaluated against objectives and specifications, and the outcome is reported. The new learning flows to the decision-making area to help improve design for subsequent projects, while the knowledge created is reflected upon within the project management area to update and improve processes to ensure that errors are not repeated. The actions taken to improve processes in the project management area are similar to the ‘act’ step of the PDCA cycle where actions are applied to the outcomes of the previous step for necessary improvement.

Further, as the project management area benefits from reflecting upon the new knowledge created and enhanced collective experience of the organisation, it transfers this knowledge to the decision making area to help improve its functioning and ensure that the correct, or improved, choices are made. This begins the next cycle of knowledge flows within the functional areas of the K-DFM, similar to Grady’s (1997) iterative PDCA cycles. The K-DFM therefore provides continuous process improvement that enables the processes to be more efficient and effective, and produce better quality software. Table 10 below presents a summary of the above discussion.

**Table 10 Evaluating K-DFM with Grady's PDCA Cycle**

No	Grady’s PDCA Cycle Step	K-DFM Functional Area and Knowledge Flows
1	<b>Plan</b> – Establish objectives and processes necessary to deliver desired results	<b>Decision-making</b> – Decide and make choices about development plans
2	<b>Do</b> – Implement the processes	<b>Technical Development</b> – Develop the product
3	<b>Check</b> – Monitor and Evaluate processes and results against specifications; report outcome	<b>Quality</b> – check for defects, correct errors, and ensure product conforms to required specifications
4	<b>Act</b> – Modify for necessary improvement	<b>Project Management</b> – Update and improve processes to ensure mistakes are not repeated within future projects

Table 10 presents an evaluation of the steps of Grady’s (1997) PDCA software improvement cycle and the functional areas and knowledge flows of the K-DFM. The K-

DFM is not limited only to the functional areas but goes beyond with the knowledge flows enabled by the feedback loops. Eskerod and Skriver (2007) state that 'the main argument in the current literature is that knowledge transfer and reflections on lessons learned may bring companies and organisations a competitive advantage that cannot be obtained by only performing project management.' The feedback loops of the K-DFM facilitate the flow of knowledge creation, transfer, application, learning, along with experience and reflection. The focus on the flow of knowledge transfer and reflection, along with the dynamic nature of the feedback loops that globalise knowledge within the software development effort, thus provide the K-DFM with the ability and potential to deliver competitive advantage.

#### **9.4.2 Assessing the K-DFM – Revisiting Rubenstein-Montano et al's Criteria for Knowledge Management Frameworks**

Section 2.5 discusses Rubenstein-Montano et al's (2000) review of knowledge management strategies and frameworks. To summarise the discussion, Rubenstein-Montano et al (2000) recommend that a knowledge management framework should:

- be both prescriptive and descriptive, that is a combination of the two approaches
- be consistent with systems thinking
- link knowledge management to organisational goals and strategies
- be planned before the knowledge management activities take place
- acknowledge the organisational culture, and the knowledge management practices must be compatible with the culture
- direct knowledge management through learning and feedback loops

The K-DFM is a dynamic model that facilitates the flow of knowledge between the functional areas of project management, decision-making, technical development, and quality, through feedback loops. The model is descriptive in its depiction of the flow of knowledge between the four functional areas. However, rather than being prescriptive, the model facilitates the flow of knowledge. The model uses a systems approach and depicts the relationships and interactions of project management, software development and knowledge management. In doing so, the K-DFM highlights the fact that consideration must be given to non-technical aspects of the software development effort. The function of the decision-making area is to integrate different perspectives and considerations, and make sense of the knowledge that is created and emerges from the functional areas and

flows through the feedback loops, thereby making the K-DFM consistent with systems thinking.

By facilitating the flow of knowledge through the feedback loops, the K-DFM provides the organisation with the ability to provide knowledge management support to its software development and project management processes. The K-DFM provides the infrastructure that facilitates the flow of knowledge and hence supports knowledge sharing activities. Thus the model provides the framework that links knowledge management to a software project organisation's goal and strategy of continuously improving its processes in order to make them more efficient, effective and productive.

As mentioned, the K-DFM gives consideration to the non-technical aspects of the software development effort. The model acknowledges the important role of organisational culture in the effectiveness of the knowledge management initiative of a software project organisation. An organisation's culture is central to encourage interaction between individuals which is important to facilitate knowledge flow, and also provides individuals the ability to self-organise their own knowledge to facilitate knowledge sharing and therefore problem solving, O'Dell and Grayson (1998). The K-DFM provides the framework to facilitate such knowledge sharing with all functional areas, a further enables knowledge and new learning to flow through the feedback loops. Therefore, assessing against Rubenstein-Montano et al's (2000) criteria for knowledge management frameworks, the K-DFM:

- is a dynamic model
- is consistent with systems thinking
- links knowledge management to a software project organisation's goal and strategy of continuous process improvement
- provides knowledge management support to project management and software development processes
- acknowledges organisational culture and provides the framework to facilitate interaction and knowledge sharing
- enables the flow of knowledge and learning through feedback loops

Thus the K-DFM appears to match and satisfy Rubenstein-Montano et al's (2000) criteria for knowledge management frameworks. Table 11 presents how the K-DFM satisfies the knowledge management framework criteria.

**Table 11 Assessing K-DFM against Knowledge Management Framework Criteria**

No	Knowledge Management Framework Criteria	K-DFM Characteristics and Features
1	Combination of prescriptive and descriptive approaches	The model is dynamic, facilitative and descriptive
2	Consistent with systems thinking	Uses a systems thinking approach
3	Link knowledge management to organisational goals and strategies	Links knowledge management to a software project organisation's goal of continuously improving its processes
4	Be planned before the knowledge management activities take place	Provides knowledge management support to project management and software development processes
5	Acknowledge the organisational culture, and the knowledge management practices must be compatible with the culture	Considers non-technical aspects of software development; knowledge activities within and between the functional areas rely upon the organisation's culture
6	Direct knowledge management through learning and feedback loops	Enables the flow of knowledge through feedback loops

### **9.4.3 Summary of Functional Areas, Knowledge Flows and K-DFM**

Software project organisations require a continuous perspective and a systems approach, to cater to their long-term knowledge and project management needs. The K-DFM provides the framework that incorporates such a perspective and facilitates the flow of knowledge between the functional areas of project management, technical development, quality and decision-making. In doing so, the K-DFM uses a systems approach and integrates the interactions of the cross-discipline perspective of knowledge and project management, and software development. The research analysed how the interactions and interdependencies between the different functional areas are integrated, and provided a complete picture of how the long-term requirements of a software project organisation can be addressed, similar to the theoretical propositions of the DFM. The research also evaluated the knowledge flows between the functional areas of the K-DFM against Grady's (1997) PDCA software improvement cycle, to assess the model's capability to improve software processes and make them more efficient, effective and productive. The research further

assesses the K-DFM against the criteria for assessment of knowledge management frameworks and strategies as suggested by Rubenstein-Montano et al (2000), which the model appears to match.

## **9.5 Conclusion**

Based upon the empirical evidence of the previous chapter, this chapter analysed how knowledge flows within the software development and project management processes of a large organisation such as XYZ. The chapter established the flow of knowledge within and between the functional areas of project management, technical development, decision-making and quality assurance. The chapter also identified the infrastructure and mechanisms that facilitate and support this flow of knowledge within the functional areas and processes. The analysis confirmed feedback and interaction between the functional areas and processes as proposed by the DFM and K-DFM. Therefore, this chapter validates the hypothesis and research question as formulated within the Dynamic Feedback Model by identifying the four functional areas and feedback loops within the operations and processes of software organisations, and that the flow of knowledge between these areas determines the effectiveness of the organisation's development activities in the long term.

## Chapter 10 KM Capability Framework

This chapter presents a Knowledge Management Capability framework based upon the empirical case study conducted at XYZ software project organisation. The chapter discusses the development of the organisation's knowledge management initiative from its initial state, to an organisational state where the KM practices are institutionalised and embedded within the daily activities and work methods of the organisation. The organisation's KM initiative is analysed through the development of two KM capabilities, namely infrastructure and processes. These KM capabilities are an integral part of the research questions developed in Chapter 5, and were examined in depth while conducting the case study. The discussion in this chapter also emerges from the categories and attributes of Table 7, and their first appearance is marked in brackets.

### 10.1 KM Capabilities

The previous chapter identified and discussed the flow of knowledge within the development processes of software projects at XYZ. The chapter also analysed the knowledge infrastructure (C3-A13) and processes that support and facilitate this knowledge flow. XYZ now possesses knowledge infrastructure such as leadership (C2-A7), top management support (C2-A4), a knowledge culture (C10), and IT capability in the form of repositories (C7-A2), asset libraries, intranet portal (C3-A16) and collaborative technology. The researcher also observed the knowledge processes of knowledge creation, storage, retrieval, transfer and application (C3) during the process of software development at XYZ. These knowledge processes manifested in the form of training (C2-A1), mentorship (C2-A6), interaction (C1), feedback (C12), and collaboration.

The knowledge management initiative at XYZ started as a concept and is now developing into a state where knowledge management practices are being increasingly institutionalised and embedded into the daily work practices and methods of the organisation. For a knowledge management initiative to achieve such an organisational state, the knowledge infrastructure and process capabilities (C3-A21) also need to develop from an initial state of low availability, accessibility, usage and practice to a state of organisational capability of high availability, accessibility, usage and practice (Gold et al 2001, Khalifa and Liu 2003). This research adopts KM infrastructure and processes as two dimensions of KM

capabilities, and the following sub-sections explain the rationale for adopting them to analyse development of KM capabilities.

### **10.1.1 KM Infrastructure Capabilities**

Gold et al (2001) identify information technology, organisational structure, and culture (C10) as infrastructure capabilities, and acquisition, conversion, application and protection as process capabilities, and Khalifa and Liu (2003) while advancing Gold et al's (2001) proposition establish leadership (C2-A7), culture and KM strategy (C3-A7) as infrastructure required to develop a knowledge management initiative.

Information technology is an infrastructure capability as it facilitates knowledge flow and eliminates barriers to communication within an organisation. A flexible organisational structure encourages knowledge sharing and collaboration across boundaries within the organisation, while a rigid structure often has the unintended consequence of inhibiting such practices. Organisational structure capability for facilitating knowledge flow is also shaped by the organisation's policies, processes, and system of rewards and incentives, which determine the channels from which knowledge is accessed and how it flows (Leonard-Barton 1995). An organisation's culture is central to encourage interaction (C1) and collaboration between individuals that are important to facilitate knowledge flow (C3), and also provides individuals the ability to self-organise their own knowledge and practice networks to facilitate solutions for problems and share knowledge (O'Dell and Grayson 1998). Organisational vision, mission and values embody the culture of the organisation and determine the types of knowledge that are desired and the types of knowledge related activities that are encouraged (Leonard-Barton 1995). Leadership sets the overall concept and implementation plan for the knowledge management initiative and obtains commitment from individuals to achieve the desired objectives and outcome. The KM leader helps create the appropriate culture to accomplish the knowledge vision and strategy of the organisation. The knowledge management strategy identifies the knowledge requirements and how they are to be fulfilled in congruence with the strategic goals of the organisation.

### **10.1.2 KM Process Capabilities**

The knowledge management processes of an organisation are focused towards obtaining, sharing, storing, and using knowledge. Examples of these aspects of knowledge management processes within the literature are: capture, transfer, and use (DeLong 1997); acquire, collaborate, integrate, experiment (Leonard-Barton 1995); create, transfer, assemble, integrate, and exploit (Teece 1998); create, transfer, use (Spender 1996, Skyrme and Amidon 1998); create, process (Ivers 1998); create, store; transfer and apply (Alavi and Leidner 2001); acquire, convert, apply, protect (Gold et al 2001). An examination of the characteristics of knowledge process capabilities enable them to be grouped into the four broad dimensions of knowledge creation, conversion, transfer and application.

Knowledge creation is enabled by the processes and activities of interaction, feedback, innovation, brainstorming, and benchmarking. Knowledge conversion is made possible through the processes and activities of synthesising, refinement, integration, combination, coordination, distribution and restructuring of knowledge. Shared contexts and common representation are required for knowledge conversion, and mechanisms for facilitating the same are group problem solving and decision-making. Information technologies like email, repositories, intranet portal, teleconferencing, and the activities of mentoring, collaboration and training play a key role in transferring knowledge. Forums such as communities of practice and centres of excellence, and training provide a platform for the transfer of knowledge. Knowledge is effectively applied during the developmental processes of an organisation through rules and directives, routines and self-organised teams. Knowledge is applied to formulate and refine the standards, procedures and processes developed to execute tasks within the organisation.

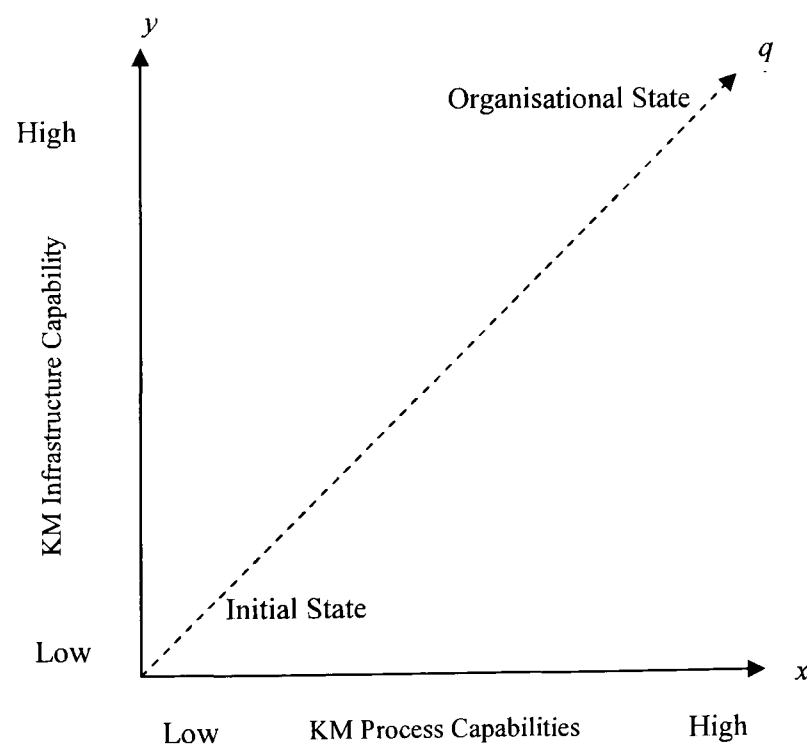
## **10.2 Development of KM Capabilities at XYZ**

The above knowledge processes are dynamic and highly interdependent and intertwined. At any point of time and in any part of an organisation, individuals and teams maybe engaged in several different aspects of these knowledge processes. The main focus of the knowledge processes is to facilitate the flow of knowledge between individuals, and consequently teams, and therefore a major challenge for any knowledge management initiative is to facilitate these flows so that the maximum amount of transfer occurs. To meet this challenge, knowledge management process capability needs be fully leveraged, and this is not possible without the presence of knowledge management infrastructure



capability. Gold et al (2001) state that ‘the presence of both knowledge management process and infrastructural capabilities is critical to reach the intended knowledge management objectives.’ Appropriate knowledge management infrastructure needs to be implemented to routinise knowledge management processes and practice and to enhance knowledge application in daily business procedures, Grant (1996).

As organisations implement knowledge management initiatives, the knowledge management infrastructure and processes develop. One might expect the development of these knowledge management infrastructure and processes would progress smoothly and in congruence with each other, from an initial state to an organisational state where the KM capabilities are embedded in the daily activities and work practices of the organisation. The path of such an ideal development is represented in Figure 15 where KM infrastructure capability development is represented on the  $y$ -axis and KM process capability is represented along the  $x$ -axis of the graph and both capabilities progress from low to high along their respective axis. The ideal, congruent development of both capabilities is represented by arrow  $q$ , which depicts a smooth progress from an initial to an organisational state.



**Figure 15 KM Capabilities**

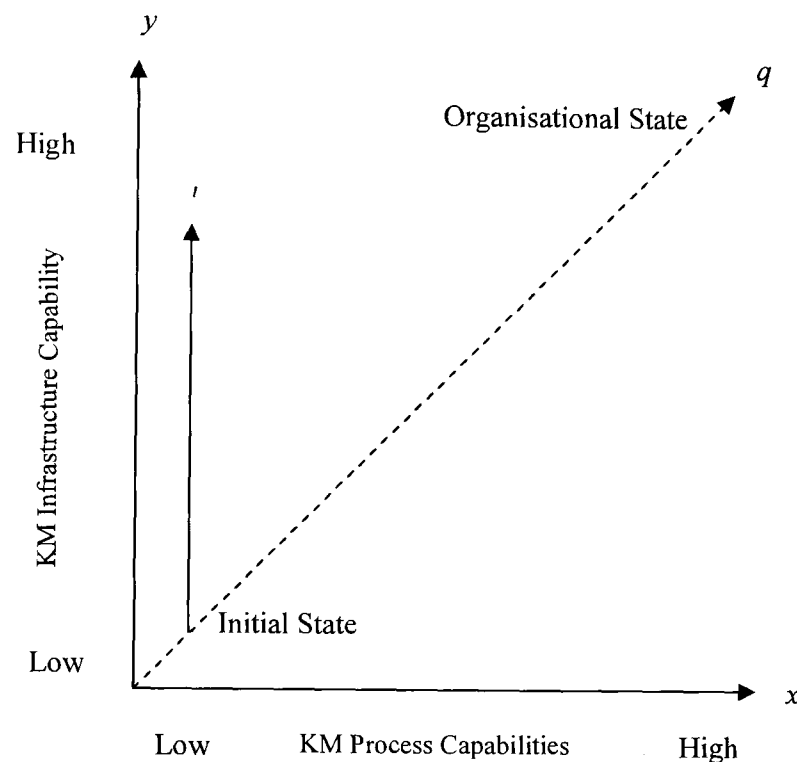
### **10.2.1 The Initial State**

However, the research observed that in actual practice at XYZ the path taken during the development of KM capabilities was not smooth and ideal, as represented by arrow  $q$ . As

mentioned previously the KM initiative at XYZ evolved from the early days of documents stored in physical libraries. Individuals were, perhaps unknowingly, performing knowledge process activities while referring to these documents and past project data, and interacting with colleagues. While XYZ always possessed leadership, the organisational structure, culture, vision, and use of collaborative technology also evolved over a period of time. The creation of a central repository marked the beginning of a determined effort to harness the use of technology to improve the efficiency and productivity of existing and future projects. The realisation of the benefits of such efforts motivated senior management at XYZ to explore further possibilities and create a knowledge vision, thus signifying the initial state of development of KM infrastructure and process capabilities. During this stage XYZ defined what KM meant to it as an organisation, and made clear the concepts and objectives that it wanted to achieve by implementing a KM initiative. A KM strategy was developed ensuring that it was connected to other organisational needs and initiatives that already existed, and resources and infrastructure required to implement the initiative were identified. This initial stage of XYZ's KM initiative was during the years of early to mid 1990s.

Thereafter, XYZ started to develop the KM initiative, and consequently the infrastructure and process capabilities. The knowledge vision was translated into action by means of mission and value statements to encourage the growth of knowledge within the organisation. A knowledge culture of sharing was promoted and individuals encouraged to contribute, while project managers were expected to lead their teams in a learning environment of openness, trust and feedback. The introduction of collaborative technology was viewed as a significant step towards establishing the knowledge culture and to a certain extent a change in the organisational structure. The use of email, teleconferencing and bulletin boards were expected to promote collaboration and boundary crossing within department and development centres and hence reduce the silo effect of a previously more vertical structure. A central repository was developed to store process assets and process improvement proposals, while the intranet portal was developed to provide organisation wide dissemination of explicit knowledge and tools such as IPMS, EKMS, HRS, and CRM. Training was imparted to introduce and make individuals explicitly familiar with these knowledge infrastructure and capabilities. Knowledge sharing activities were made mandatory within the training programmes. However the emphasis on developing KM infrastructure capability while still providing training in knowledge process capability, resulted in high availability of this infrastructure to individuals within the organisation and is represented by arrow *i* in Figure 16 Arrow *i* depicts the actual progress of XYZ's KM

initiative development contrary to the expectation depicted in Figure 15, from a state of low infrastructure and process capability to a state where the emphasis on infrastructure capability development was greater than the practice of knowledge process capability. In other words, this state was characterised with a high availability and accessibility of infrastructure capability for individuals compared to the extent to which they were performing KM processes. This stage of XYZ's KM initiative can be related to the years of the late 1990s and the early years of the next decade.



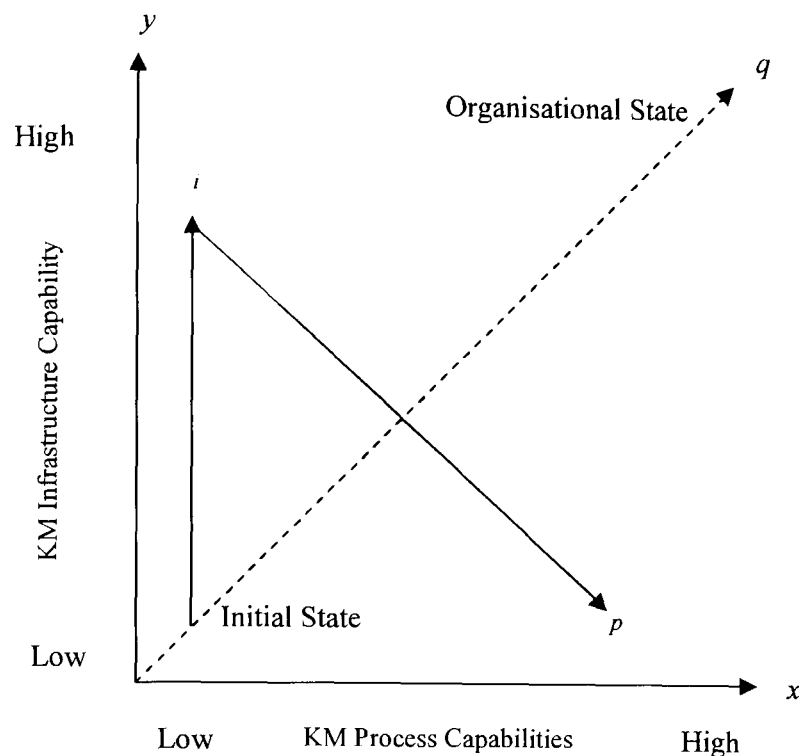
**Figure 16 KM Infrastructure Development**

### 10.2.2 The Deviation

When the researcher first visited XYZ and commenced research the number of individuals employed by XYZ was 50,000. Thereafter, the researcher made numerous visits to XYZ over the course of the next two years and the number of individuals employed by XYZ rose to 85,000. In an interview conducted in February 2007, a senior Group Lead at XYZ stated that “fifty percent of our new employees have been at XYZ for less than three years.” Therefore the number of individuals at XYZ had grown from 40,000-45,000 to 85,000 in two years. The increase in the number of new employees was representative of XYZ's expansion strategy which was characterised with the acquisition and opening of development and delivery centres across the globe. This resulted in XYZ becoming a larger global organisation with employees from diverse backgrounds and cultures working in a more distributed environment. A small number of new employees were recruited as part of XYZ's strategy to employ *“bench strength that would provide a bigger talent*

*pool.*” The idea of employing ‘bench strength’ (C10-A7) was that XYZ would provide individuals ongoing training and therefore have reserve skilled employee resources for job rotation, cover for absentees, and starting new projects. However, the number of individuals employed as ‘bench strength’ was less than five percent of the total number of new employees.

The rapid expansion had an effect on XYZ’s KM initiative, and the infrastructure and process capabilities. New employees were provided with training to perform knowledge process capabilities as a part of their induction programme. However, when they were assigned to projects upon completing their training, the infrastructure capability proved inadequate and insufficient. There was a loss of knowledge richness due to the distributed knowledge, less face-to-face interaction, and scattered knowledge assets. There was a perceived lack of ‘teamness’ that resulted in a coordination breakdown in project management activities. The large numbers of new employees, or inductees, were not exposed and used to the organisational culture, and therefore did not have the requisite level of trust and familiarisation to integrate with the knowledge management practices within the organisation. Similarly XYZ had to address cultural differences amongst globally distributed employees who had to adjust to new work practices. While the new employees’ initial training helped overcome some of these issues and inculcated knowledge processes, the infrastructure capabilities of organisational structure, culture, information technology and KM strategy needed to be reassessed and improved upon. This resulted in XYZ’s possessing inadequate KM infrastructure capability compared to the number of employees seeking to perform KM process capabilities. The progression to this state is depicted in Figure 17 by arrow *p* from the previous state of high infrastructure capability and low process capability, to a new state of low infrastructure capability and high process capability.

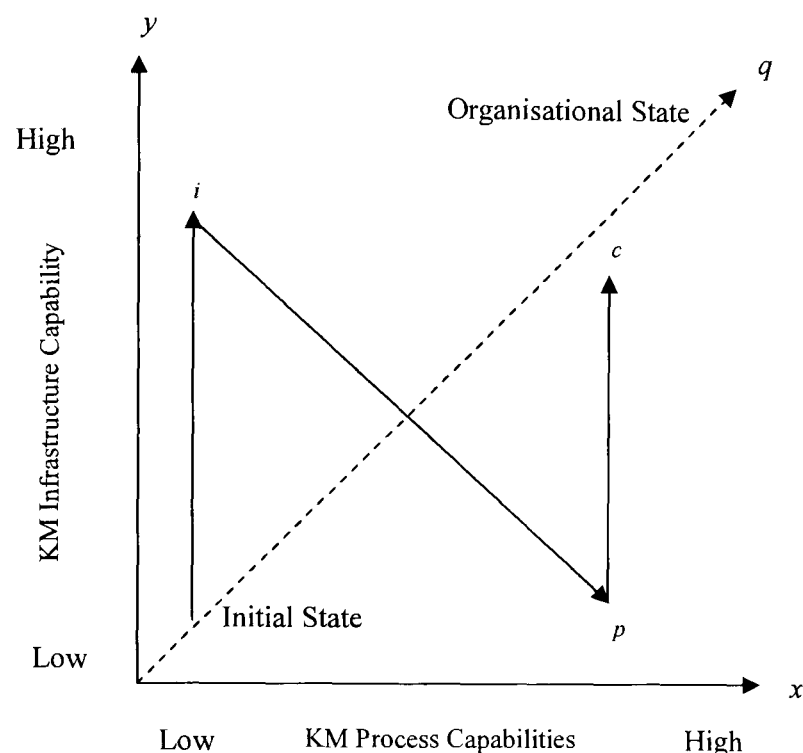


**Figure 17 KM Process Capability Development**

### 10.2.3 The Correction

Having recognised the problem XYZ addressed the issues presented by the state of low infrastructure capability and high process capability by increasing site visits and travel of individuals amongst different development and delivery centres, having local, acculturated knowledge champions and interaction among them at the regional and corporate level, encouraging regional and virtual communities of practice, and also starting regional and corporate centres of excellence. XYZ attempted to create a combination of both “top-down and bottom-up knowledge culture.” The knowledge champions were made responsible of ensuring that knowledge created at the global development and delivery centres, was made available to the local or regional centres and the overall knowledge owner at the corporate level. The existing EKMS was upgraded to a new knowledge management system named DEF which as mentioned by a senior group lead during an interview. *“consolidated all scattered knowledge assets into one system that caters to the global needs of 85,000 diverse employees,”* (the interview was conducted when the number of individuals employed at XYZ was 85000). The KM-P was introduced as an integral part of DEF to help identify experts and individuals with experience for projects with specific characteristics. DEF was implemented in late 2006, and along with the other measures mentioned above, was expected to be a catalyst that drives knowledge flow, and progresses XYZ to an organisational state of high infrastructure and process capability, where KM

practices are institutionalised and embedded in the daily activities and processes of the organisation. This progress of XYZ's KM initiative from a state of low infrastructure capability and high process capability to a state high infrastructure and process capability is depicted by arrow *e* in Figure18 completing the N-shaped journey to a higher organisational state in contrast with the initial expectation of smooth transition.



**Figure 18 Towards an Organisational State of KM Capability Development**

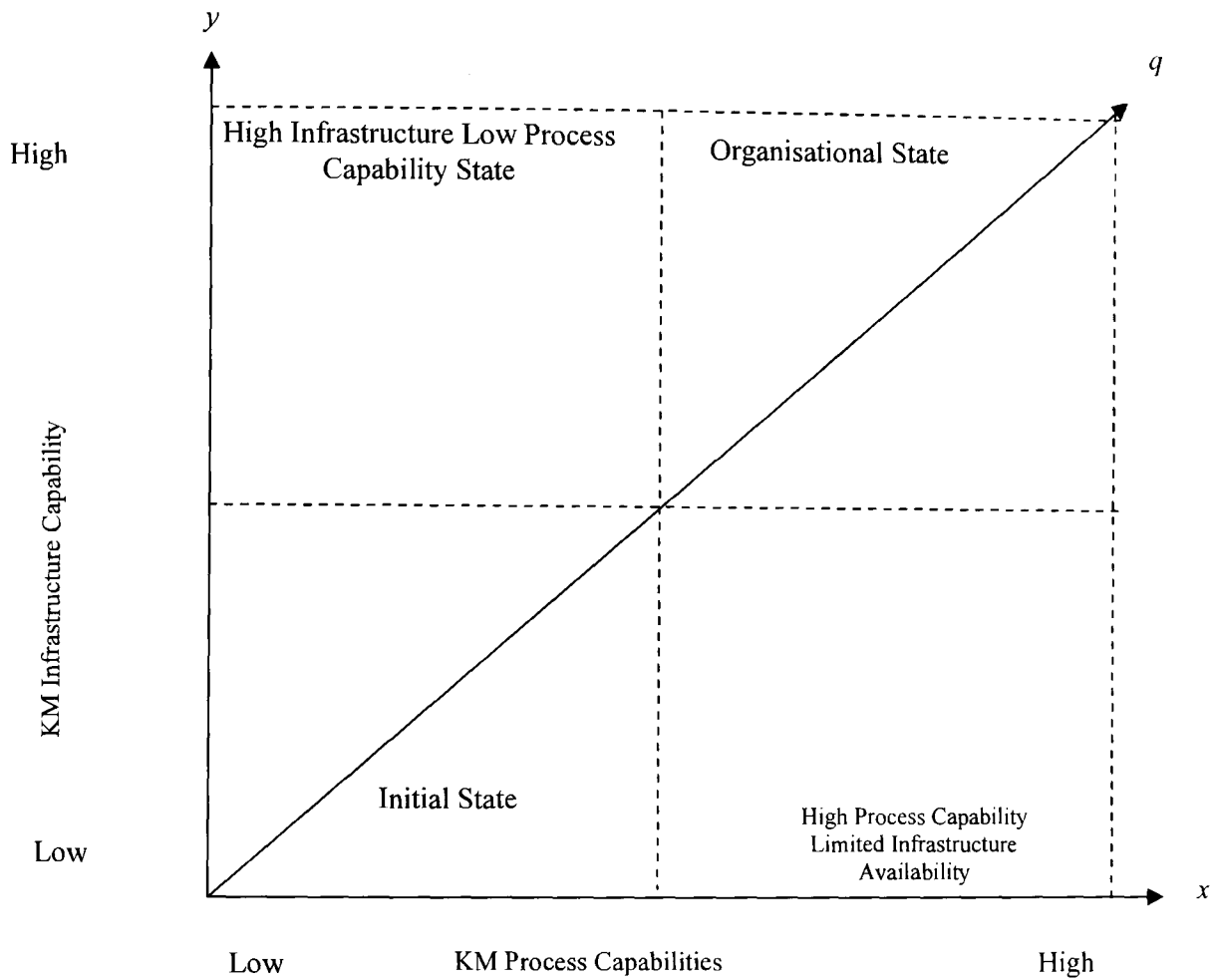
The implementation of the KM initiative at XYZ provides an example of how organisations need to balance the growth and development of KM processes and infrastructure while developing their KM programmes. Organisations expect a smooth path from conceptualising a KM initiative to its successful implementation. XYZ's experience highlights two stages within the implementation of its KM initiative when an imbalance existed between the KM infrastructure and process capabilities. When XYZ was developing the concept of the KM initiative after its initial conceptual stage, the organisation put an emphasis on developing the infrastructure. This resulted in greater availability of KM infrastructure capability than KM processes being practiced, even though training was introduced for these processes, thereby representing a state of higher KM infrastructure capability and lesser KM process capability. Thereafter, with the addition of a number of employees and their training upon induction, the KM infrastructure capability was inadequate to support the KM processes practiced by the individuals. This represented a state of greater KM processes being practiced and lesser KM infrastructure capability being available. XYZ started progressing towards a state of organisational KM capability after it addressed these imbalances.

### 10.3 KM Capability Framework

The above discussion highlights the issues faced by XYZ while developing a knowledge management initiative in order to mobilise and utilise its knowledge resources. While the findings pertain to a single organisation, they may also reflect the view of Eskerod and Skriver (2007) who studied the literature on knowledge transfer and identified persistent issues that impact such efforts suggesting a more general trend. Eskerod and Skriver (2007) state that ‘however.....many companies experience serious problems when trying to make knowledge transfer work.’ In the case of XYZ, the organisation struggled to make knowledge resources available to all individuals when it inducted a significant number of new employees. This problem is made apparent by the downward arrow *p* in Figures 17 and 18 when the KM infrastructure was found inadequate to support the knowledge needs of a larger number of organisational individuals. Thus Eskerod and Skriver’s (2007) view helps explain the phenomenon observed at XYZ.

The discussion in Section 10.2 also confirmed that if an organisation conceptualises its KM programme in an initial state and intends to achieve an organisational state where the KM capabilities are institutionalised and embedded within the organisation’s daily procedures, processes and practices, two other intermediate and distinct capability states also exist. One state is of higher KM infrastructure capability availability and lesser KM process being practiced, while the other state is of greater KM processes being practiced and lesser KM infrastructure capability being available. The four states are represented in Figure 19. Also represented in the figure is the ideal path an organisation would expect to progress along when launching a KM programme, and indirectly the possible paths along which their KM programme might progress during implementation. The path to implementing a KM programme does not progress directly and smoothly from the initial to organisational state as envisaged by XYZ, but might instead progress through either of the two intermediate states, or indeed as in XYZ’s case through both intermediate states. If an organisation initially lays more emphasis on developing its KM infrastructure capability it will progress to the state of higher KM infrastructure capability and lesser process capability before it can progress to an organisational state. On the other hand, if the organisation was to initially lay more emphasis on practicing KM processes it will progress to the state of greater process capability and lesser infrastructure capability and before being able to progress towards the organisational state. However, as XYZ’s experience depicted, a large organisation could progress from one intermediate state to another before progressing towards the organisational state of KM capability. This is a very important observation

because many organisations tend to launch knowledge management programmes without due consideration of the organisation's capabilities to guarantee any measure of success of implementation (Davenport and Prusak 1998, Leonard-Barton 1995). As the case study evidence revealed, even one of the largest software project organisation with CMMI Level 5 accreditation, needed to coordinate its KM capability development to achieve a state of organisational knowledge management.



**Figure 19 KM Capability Framework**

The framework presented in Figure 19 depicts the possible states organisations may progress along while implementing their KM programmes. In the following sections the research discusses the characteristics of each state, and the possible stage of an organisation's KM programme development.

### 10.3.1 Initial State

An organisation's KM programme can be considered to be in the initial state when the organisation is creating a knowledge vision and relating this vision to its strategic needs and other initiatives that already exist. During this state the organisation explores all



possibilities related to the KM initiative and also the opportunities present. The organisation identifies the infrastructure required to support the initiative and the KM processes to be practiced. Financial support for the programme and other resources required to implement the programme are also identified and budgeted. An important activity or feature of this state is the top management's commitment to the KM initiative and development of a cross-functional team responsible to implement the programme. Within this state, management needs to communicate its knowledge vision across the organisation, and make individuals aware to the KM programme and its expected benefits.

### **10.3.2 High KM Infrastructure Capability**

The KM programme is in the state of high infrastructure capability when there is an emphasis on developing the infrastructure. During this stage the knowledge vision is translated into action by means of mission and value statements to encourage the growth of knowledge within the organisation. A knowledge culture of sharing and learning is promoted with individuals encouraged to participate and contribute. The organisation reviews its policies and processes, and implements systems of rewards and incentives to motivate and reward knowledge sharing behaviour. During this state information technology support is developed in the form repositories and collaborative technologies. Through the linkage provided by collaborative technologies the organisation attempts to integrate previously fragmented flows of knowledge, Teece (1998). Collaboration technologies are developed to allow individuals within the organisation to collaborate, thereby eliminating the structural and geographical impediments that may have previously prevented such interaction. Knowledge discovery technologies are developed to allow the organisation to find new knowledge that is either internal or external to the firm. Knowledge mapping and application technologies are developed to enable the firm to effectively track sources of knowledge, creating a catalogue of internal organisational knowledge, and apply its existing knowledge. An organisation's KM programme could be considered to be in this state when individuals have access to the above mentioned infrastructure but do not avail themselves of its complete potential or capability, due to the lack of practicing knowledge processes.

### **10.3.3 High KM Process Capability**

The KM programme can be considered to be in the state of high KM process capability when there is an emphasis on practicing knowledge processes. Openness and trust characterise the organisation's work environment and support knowledge sharing behaviours, which are included as an integral part of the training programmes. Communities of practice and centres of excellence evolve and individuals are encouraged to join and participate. Activities that establish an organisation's KM programme in a state of high knowledge process capability include identifying lessons learnt, best practices, benchmarking, brainstorming, group problem solving, mentoring and collaboration. The daily work processes support decision-making, feedback and interaction, which are made apparent in the team commitment. Knowledge champions from distributed centres meet regularly and knowledge flows across boundaries and development centres. Therefore an organisation would be in a state of high knowledge process capability and low infrastructure capability when the above mentioned knowledge processes are practiced but do not receive adequate support in the form of infrastructure support.

### **10.3.4 Organisational State of KM**

An organisation will be in a state of organisational KM infrastructure and process capability when it achieves high availability of infrastructure capability to support frequent and regular practice of knowledge processes. In other words, knowledge processes are embedded in the daily routines, procedures and practices of the organisation which possess the knowledge infrastructure to support them. This state is characterised by a vibrant mix of vision, strategy, leadership, organisational structure, culture, technology infrastructure, and knowledge processes of creation, storage, retrieval, transfer, application and sharing. Forums such as communities of practice evolve and the organisational structure, culture, and technology support them. Lessons learnt are captured regularly and made available across the organisation, while best practices are implemented. Knowledge sharing and learning permeate the organisational environment of role models, mentoring, leadership, motivation, commitment, and training, where collaboration, feedback and interaction drive knowledge flow between individuals and teams. Acculturated knowledge champions and collaborative information technology support ensure that knowledge flows are not inhibited by organisational structures and distributed geographical locations, but instead flow across social networks and boundaries of the organisation. The knowledge flows ensure that the knowledge available within the organisation is current, integrated, usable.

and applied. The organisation adopts a consistent approach to KM and it becomes a way of working within standardised work methods. Thus when KM is institutionalised within the organisation, the programme can be stated to be in the organisational state.

### **10.3.5 Framework Summary**

The above discussion presents the activities and characteristics of the states through which the implementation of organisational KM programmes could possibly progress. By assessing and focusing on the KM infrastructure and process capabilities and their characteristics that are being developed and practiced, organisations can determine the current state of their KM programme implementation. The framework enables organisations to analyse if their KM programme is more focused towards developing KM infrastructure capability rather than KM process capability, or whether limited KM infrastructure is available for the KM processes being practiced. Thus, the framework helps organisations to better understand the issues related to developing a KM initiative, as suggested by Eskerod and Skriver (2007), and analyse any imbalance that may exist and needs to be addressed. In doing so, the framework benefits organisations interested in making corrections and restoring the balance between KM infrastructure and process capability, thereby smoothing the path of successful KM implementation towards a state of organisational KM capability.

## **10.4 Conclusion**

The case study analysed knowledge flows during the software development and project management processes, and feedback and interaction between and within the functional areas of the operations of a software project organisation. The case studied also identified the knowledge management infrastructure and process capabilities required to support and facilitate this knowledge flow. This chapter further analysed the development of these KM infrastructure and process capabilities from an initial state to an organisational state. Not all organisations will manage to progress to the organisational state of KM in one smooth journey, as observed in the XYZ case study. The chapter established that two other intermediate states exist, and identified the possible paths an organisation's KM capabilities development might progress along.

## Part IV Conclusions

## **Chapter 11 Conclusions**

This research investigated how experience from executing earlier projects can be applied for more effective implementation of future projects. Experience includes the existing knowledge of individuals, new knowledge created during team interactions, insights gained and lessons learnt while executing projects. Therefore the research analysed how knowledge generated while implementing software projects benefits subsequent projects. Feedback and interaction are the key elements that facilitate learning and knowledge creation. Chapter 5 presented a framework that provides feedback and interaction between the functional areas of project management, decision-making, technical development and quality assurance. Feedback and interaction thus facilitate the flow of knowledge within these functional areas of a project and provide a continuous view of software development. The continuous view provides an organisation with the long-term perspective of applying knowledge to future software projects. This research conducted a case study to examine the flow of knowledge, and the capabilities required to support this flow, within and between implementation projects of a large international software development organisation. The findings of the case study validated the flow of tacit and explicit knowledge within the functional areas of a project, and subsequent projects, as proposed in the extended K-DFM model, and further motivated the research to present a framework that will benefit organisations developing KM capabilities while implementing a KM initiative. This chapter concludes and summarises the main findings, contributions and implications of this thesis, and discusses the possibilities and scope of future work based upon the outcomes of this research.

### **11.1 Key Points**

The research propositions required a study of how a software project organisation's knowledge management initiative supports project management and software development processes. The research conducted a case study at a large CMMI Level 5 software project organisation that has over 150 offices worldwide in more than 40 countries and employs in excess of 108,000 individuals. Exceptional access to the organisation provided the research with an opportunity to study and analyse the well established and highly mature work methods and practices of the software development effort. The following were the key findings and outcomes of the study.

### 11.1.1 The Knowledge Dynamic Feedback Model (K-DFM)

The case study provided evidence of the flow of knowledge and how individuals work while implementing software projects. The research identified that software organisations broadly categorise their development activities within the functional areas of project management, technical development, quality and decision-making, and knowledge flows within and between these areas. This confirmed the premise presented within the DFM, a theoretical published model, of the four functional areas and their feedback loops. The research adapted the model by analysing the possible flow of knowledge while implementing projects in a software project organisation. The research validated the adapted version of the DFM by identifying the four functional areas and the flow of knowledge, and established that software organisations implicitly design their operations and work practices around these functional areas. The research examined how knowledge, both tacit and explicit, flowed between the functional areas of the organisation's software development operations. Further analysis of the interactions and relationships of the functional areas and knowledge flows determined the feedback loops. Thus the research was able to establish *how* knowledge flowed in a dynamic manner at the software project organisation. This resulted in a Knowledge Dynamic Feedback Model (K-DFM).

The K-DFM was evaluated against Grady's (1997) PDCA software process improvement cycle. The activities within the K-DFM's functional areas of decision-making, technical development, quality and project management were mapped with the plan-do-check-act steps of Grady's (1997) cycle. Therefore similar to Grady's (1997) PDCA cycle, the dynamic and iterative flow of knowledge within the functional areas of the K-DFM helps improve the organisation's effectiveness while implementing software development projects. This knowledge flow provides the software organisation with the continuous and long-term perspective of learning from past projects and applying that experience to implement future projects more effectively.

The K-DFM was also assessed against Rubenstein-Montano et al's (2000) criteria for knowledge management frameworks. The model satisfied the criteria which established that the K-DFM: provides a facilitative and descriptive approach; is consistent with systems thinking; links knowledge management to a software project organisation's goal and strategy of continuous process improvement; provides knowledge management support to project management and software development processes; acknowledges

organisational culture and provides the framework to facilitate interaction and knowledge sharing; and, enables the flow of knowledge and learning through feedback loops.

### **11.1.2 Directory of Experts**

Decision-making supports and drives the activities of problem solving within project management, technical development and quality. The effective implementation of software projects depends upon the individuals who execute them, and these individuals while working in teams rely upon their competencies, creativity, decision-making and problem solving skills. Such skills are developed through training and experiences gained while implementing projects, and are important to determine the roles and responsibilities of individual team members within project teams. On an organisational scale, an important feature of any KM programme is to ensure that individual expertise is made available across the organisation for the effective implementation of projects. This feature becomes more critical in the case of globally distributed organisations, and the organisation where the study was conducted was a typical example. The organisation implemented an organisation-wide directory within its intranet to locate experts. The system was initially started with voluntary registration for individuals to make themselves available for inquiries on certain topics. However, due to limited registration the organisation, upon recommendation by the researcher, linked registration with the individual's expertise, skill and competence levels within the HR system. This mandatory registration has made it easier for individuals to locate experts across the organisation, and the experts are now required to respond to queries and help others in areas of their expertise. Further, the activities of seeking and sharing knowledge are now also linked to the organisation's appraisal and rewards schemes.

### **11.1.3 Knowledge Champions in a Globally Distributed Software Organisation**

The globally distributed characteristic of the organisation and its centralised KM programme resulted in the organisation having a KM initiative that resembled a client-server analogy. Knowledge from distributed centres would flow towards the base head-offices, and then out towards other distributed centres, resulting in a loss of knowledge richness especially in the case of distributed centres that were closely located geographically. The researcher recommended greater interaction between such closely located geographical centres, and that has now resulted in local, acculturated knowledge

champions who meet more frequently at a regional level, and regularly at a corporate or organisational level. Individual visits between these close proximity centres have also increased, resulting in more localised social networks and communities of practice at the regional level. The above characteristic highlighted the issue that not all globally distributed organisations implement a distributed KM programme. However such organisations require establishing distributed KM programmes that represent a peer-to-peer analogy, where local networks form at the regional level and also interact at the organisational level. Such a distributed KM programme is sensitive to the unique local culture and provides a degree of autonomy at the regional level, while the interaction at the organisational level ensures the degree of standardised work required. Also, the distributed KM programme and its peer-to-peer approach recognise the tacit dimension of knowledge and assume that knowledge is shared mainly through direct person to person contact. This signifies the personalisation aspect of the organisation's hybrid knowledge strategy, and complements the codification aspect that already exists with the client-server approach of amalgamating individual knowledge within a cohesive context and making it centrally available to members of the organisation via repositories and the intranet portal.

#### **11.1.4 Developing Knowledge Management Capabilities**

The problems encountered by the organisation while attempting to expand and increase the scale of their KM programme, provided the research with an insight into the issues involved while developing KM capabilities. As organisations implement knowledge management initiatives, the knowledge management infrastructure and processes develop. The development of these knowledge management infrastructure and processes are expected to progress smoothly and complement each other, from an initial state to an organisational state where the KM capabilities are embedded in the daily activities and work practices of the organisation. The implementation of the KM initiative at XYZ provides an example of how organisations need to balance the growth and development of KM processes and infrastructure while developing their KM programmes. The research analysed that as XYZ expanded as an organisation and employed more individuals, the KM infrastructure capability development did not progress at the same rapid pace of development of KM process capability. An added number of individuals were employed and trained to practice knowledge processes but the KM infrastructure proved to be inadequate. Thus as XYZ scaled up its operations and practices it needed to address the issues presented by the same.



The research mapped and analysed XYZ's progress through this KM implementation of scaling up the KM capabilities and related issues. The analysis resulted in a KM Capability framework that presents the progress of an organisation's KM infrastructure and process capability development. The framework enables organisations to assess and focus on the KM capabilities and their characteristics that are being developed and practiced, and thus determine the current state of their KM programme implementation. Organisations can analyse if their KM programme is more focused towards developing KM infrastructure capability rather than KM process capability, or whether limited KM infrastructure is available for the KM processes being practiced. Thus, the framework helps organisations to better understand the development of its KM initiative, and analyse any imbalance that may exist and needs to be addressed. Organisations can benefit from the framework and make corrections and restore balance between KM infrastructure and process capability, and therefore progress towards successful KM implementation and a state of organisational KM capability.

#### **11.1.5 Summary of Key Points**

The research investigated and evaluated the knowledge management support required to implement projects in software project-based organisations. While conducting the case study, the research examined, evaluated and analysed the knowledge management, project management, and software development processes and activities within a leading and reputed software project organisation. The outcome resulted in the presentation and validation of a feedback model that facilitates the flow of knowledge within software projects, and a framework that benefits organisations while developing knowledge management capabilities, while also helping the case study organisation to implement knowledge based solutions to certain practical problems.

Reich (2007) states that '.....many, if not all, organisational outputs can be usefully seen in knowledge-based terms, so we need to use knowledge-based concepts and theories to understand how they are produced.' Reich (2007) further states that '.....the structure and processes of organisations and the people and systems within them can be usefully understood in knowledge-based terms.' This research focused on the knowledge based concepts to understand how they support project management and software development processes to enhance and make them more effective. In doing so, the research was able to evaluate the benefits of adopting a knowledge based perspective for both project management and software development. As confirmed by Reich's (2007) view, recent

thinking within project management literature is coming to accept the benefits of the knowledge based view, and hence the value of this research can be fully appreciated by the disciplines of software development and project management.

## **11.2 Contribution**

This research investigated how knowledge generated while implementing a software development project can be leveraged and effectively reused in future software projects. The research therefore studied, analysed, and synthesised the literature, theories, principles and processes from the disciplines of knowledge management, information systems, software engineering and project management. The conceptual synthesis of this analysis presents a new perspective about the relationship and interactions between these disciplines and how knowledge management processes support software development and project management processes. A KM perspective to software development provides the continuous and long term view of harnessing and leveraging knowledge and learning to implement subsequent software projects more effectively and efficiently. Also, during the initial stages of early software projects, an emphasis is required on software processes and their execution, but as organisations progress towards implementing projects regularly, a project management perspective assumes greater significance. Consequently, the project management and software processes are further enhanced and become more effective as the organisation's knowledge management support capabilities develop.

The research was able to validate the above synthesis of knowledge management, software development and project management while conducting a case study at a large CMMI Level 5 organisation that currently employs more than 108,000 individuals. Further insights gained during the case study added to the conceptual analysis and synthesis. On the basis of the theoretical background gained through literature review and analysis of the insights gained, the researcher was able to make suggestions and solve practical problems the organisation faced while implementing its KM programme. The researcher recommended to the organisation a possible employee rotation not limited only to project implementation but also temporarily co-locating team members, during project start-up and periodically through site visits and travel. This recommendation was made primarily to address the problem of the organisation's centralised KM programme. The organisation's approach had created an emphasis on codifying knowledge and making it available centrally. The researcher's suggestion was an attempt to increase greater individual interaction and restore balance to the organisation's desired KM hybrid strategy with a

more personalised approach. Further suggestions to enhance interaction included the concept of local knowledge champions at the regional level, and designated liaisons, acculturated and informed about technical issues, as focal points of co-ordination and who can transfer knowledge to other sites through regular travel visits.

The organisation was also having a problem getting experts to voluntarily register on its 'Yellow Pages' knowledge 'map'. During the study the researcher recommended that the organisation link an individual's registration with their expertise, skill and competence levels that already existed within the HR system. The expert registration would also be updated as when an individual's profile is updated within the HR system. The organisation therefore now has available the current status of expert availability corresponding to their respective skill and competency profile, and the experts respond to queries and help others in areas of their expertise. This has resulted in organisation-wide availability of experts on a globally distributed scale.

This research presents a dynamic feedback model (DFM) that facilitates feedback and interaction which are key prerequisites for learning and knowledge creation. The model depicts the software processes as a continuous description of relationships, interactions and feedback. The research adapts and upgrades the model to incorporate the knowledge flows that underpin and support the project management and software processes. The adapted model provides the framework required to facilitate the dynamic flow and application of knowledge between the functional areas of project management, technical development, decision-making and quality assurance. The research identified these broad functional areas and the flow of knowledge while conducting the case study, and thus validated the model. The validated model (K-DFM) was further evaluated with the plan-do-check-act (PDCA) cycle of software process improvement to assess its ability to improve the effectiveness of the development effort. The K-DFM was also assessed against the criteria for knowledge management frameworks, which the model satisfied. Thus the K-DFM provides a continuous view and improvement of software development processes and incorporates the support of knowledge management processes.

The research presents an original framework to assess the development of KM capabilities within an organisation. The framework depicts the possible paths of the development of KM capabilities from an initial to an organisational state, and enables the organisation to determine the current state of its KM programme implementation. The framework helps organisations to better understand the development of their KM infrastructure and process

capabilities while implementing their KM initiative, and analyse any imbalance that may exist between them. Organisations can thus make corrections and restore balance between these KM capabilities development to ensure successful KM implementation towards a state of organisational KM capability.

The work done during the course of this research has resulted in seven refereed and published papers that were presented at international conferences. The papers analysed the theoretical concepts and issues discussed in this thesis, reported on the case study findings, and evaluated the implications of this research. The main themes of the research papers included: evaluating the DFM model against different software development life cycle models, namely, the Waterfall, the Spiral, and Rapid Application Development, and agile approaches; the uniqueness, novelty and special needs of projects; decision-making and negotiation in projects; the need to manage and the management of knowledge within projects, including knowledge creation, transfer and application; and, the development of knowledge management infrastructure and process capabilities. The refereed and published papers are listed in Appendix 3 along with their respective abstracts. Table 12 below summarises the main contributions of the thesis.

**Table 12 Contribution of Thesis**

	<b>Contribution</b>
1	<b>Synthesis</b> of theories, principles, and processes of Knowledge Management, Software Engineering, and Project Management
2	Practical Problem 1 – Recommendation to the organisation to have acculturated, local knowledge champions at regional centres, resulting in increased interaction, and greater exchange and flow of tacit knowledge, at the regional level.
3	Practical Problem 2 – Recommendation to link individual registration with expertise, skill and competence levels within HR system. resulting in organisation-wide availability of experts on a globally distributed scale.
4	<b>Development of a Knowledge Flow Model</b> Presentation of a previously published theoretical model; adaptation of the model to incorporate knowledge flows; validation of the adapted model; evaluation of validated model in comparison with Grady’s PDCA cycle of software process improvement; assessment of the validated model against Rubenstein-Montano et al’s Criteria for Knowledge Management Frameworks
5	<b>Knowledge Management Capability Framework</b> Presentation of an original framework to assess the development of Knowledge Management Infrastructure and Process Capabilities

### **11.3 Limitations**

The case study was conducted within a large software organisation where exceptional access had been granted to the researcher by top management. Senior managers at the organisation ensured that access was provided and the researcher was able to observe team meetings, interactions and processes, and conduct interactive, open-ended interviews. The researcher was able to interview senior managers from the SEPG, consultants, group leads, project managers, project leads, teams and individuals.

However there were certain limitations while conducting the case study. Being one of the largest and best reputed software organisations in the world implied that the organisation was also implementing projects of a sensitive and critical nature for a diverse range of customers. For example, an organisational facility visited frequently by the researcher

during the course of the study stored large volumes of important data and information for one of the world's leading banks, and also a large construction company, a service provider, and an airline. Therefore the security measures across the organisation were very strict, and included not permitting the use of any electronic device by a person who was not a permanent employee or member of the organisation. Even organisational employees required prior permission and authorisation if they were to be in possession of a personal laptop or data storage device. The researcher was allowed to carry only a diary to make notes, and was not permitted to record observations and interviews electronically, either in audio or video. This resulted in the researcher having to rely upon the written notes and immediate recall of observations and interviews and the use of established protocols to guarantee the value and accuracy of the data and observations. The researcher addressed this by transcribing notes on the same-day of conducting observations and interviews and following established procedures and conventions.

Though the researcher was allowed to conduct observations and interviews, he was not provided access to documentation of past projects and other historical data of projects implemented due to their sensitive nature. This limitation was significant as the researcher was unable to analyse how effectively the organisation was currently implementing projects as compared to past projects when KM capabilities did not exist or were in an initial stage of development. The researcher addressed this limitation by attempting to reconstruct past project implementation during interviews with project managers who had been implementing projects for the organisation since the previous decade. Therefore the research analysis is based upon observations and interviews of current ongoing projects, and data and stories related to past projects being provided by experienced project managers with a history of implementing projects at the organisation.

Another limitation encountered by the researcher while conducting the case study was because of the organisation's security procedures, no informal social interaction was possible with organisational employees. All interviews were pre-scheduled and if the researcher needed a clarification, a follow-up interview was required to be re-scheduled, which could take a few days, or in some cases even two weeks, to be arranged. The researcher was provided all facilities, including refreshments, in the interview room which therefore limited informal interaction, and this lack of informal discussion and interaction resulted in the data being collected in a structured and methodical manner but possibly missing out on some other interesting aspects. Certain aspects of the limitations of the research are discussed in Section 11.5, and extensions are proposed in future work.

## 11.4 Implications

This research provides a new perspective and better understanding of the interactions and relationship between software development, project management and knowledge management processes within the work practices of a modern day software organisation. Software projects are characterised by high levels of uncertainty and ambiguity and require dynamic resolution approaches. Such projects require creativity, heuristics and knowledge that is tacit, incomplete and ephemeral in addition to explicit and kernel knowledge processes. Software project organisations need to adopt a continuous and long-term perspective to implementing projects and developing their knowledge practices. This research establishes that software organisations implicitly design and organise their work practices and activities within the four functional areas of projects management, technical development, decision-making and quality assurance. The research presents a dynamic model that provides a continuous and long-term perspective of software development, and depicts the flow of knowledge within and between these four functional areas. The research identifies *how* knowledge flows while implementing a project within a software project organisation. The research therefore also establishes the processual nature of knowledge that exists throughout the organisation and is not located at one single time or space.

The above mentioned processual nature of knowledge and its flow has implications for a large part of knowledge management literature that focuses on how to make knowledge more manageable. Managing knowledge provides a connotation of control and ownership where the first step is to establish its ownership. However, it is difficult to assign ownership, and store and retrieve something that is abstract and elusive in nature. Knowledge is considered tacit by nature (Kreiner, 2002), that is, implied and understood implicitly in the situation, without being definable and visible. Capturing tacit knowledge is viewed as a challenge by software organisations that need to spread knowledge throughout the organisation for greater innovation. However, as established by this research, a possible approach is to facilitate the flow of knowledge to areas and activities where it is required and applied within a context. The approach considers knowledge as something that is made resourceful by being competently mobilised and utilised, and consequently new knowledge is created by improving the ability to facilitate, mobilise and utilise existing knowledge. For software project organisations this research has implications for their ability to manage context, provide feedback and facilitate interaction, and therefore build upon their existing knowledge resources. The research provides such

organisations with a perspective that would help them achieve excellence not only through optimisation of software process improvement of Level 5 of the CMMI, but also through knowledge creation, sharing and learning of the K-DFM. The K-DFM's focus on facilitating the flow of knowledge, learning, experience and reflection within the functional areas provides project-based organisations with the benefits of continuous software process improvements and competitive advantage.

The research examined and analysed the KM capabilities required to support and facilitate the flow of knowledge. The research mapped the KM capabilities along the two dimensions of infrastructure and processes. The resultant KM Capability framework enables software project organisations to monitor their KM capability development. Organisations can determine the progress of their capability development with relation to the sectors of the framework and ensure a balanced and successful implementation of KM capabilities. The research therefore provides a framework that addresses the development of KM *capabilities*. The research assesses the four states of KM capability development, and identifies the possible paths along which an organisation's KM capability development may progress. The research defines the characteristics that determine the state of an organisations KM capability development programme. The framework determines any imbalance in the KM capability development programme that enables corrective action, if required. While most KM frameworks discuss knowledge processes, this research presents a framework that addresses the development of both KM capabilities of infrastructure and processes, and maps them against each other.

Also, it is increasingly noted in knowledge management literature that technology is not the only factor of a successful knowledge management programme. The results of the analysis of this research provide additional evidence that organisations must develop capabilities to support and facilitate interaction and feedback to ensure the flow of knowledge. Organisations need to take steps to bring together individuals with common interests to improve their likelihood of success in knowledge sharing. Encouragement and facilitation of organisation-wide communities of practice are a positive step towards effective knowledge management, especially within a distributed organisation. This encouragement and facilitation of communities of practice that span the entire organisation can be an important component of an overall knowledge management strategy.

The success of acculturated regional knowledge champions and frequent interaction at a regional level of a distributed organisation also reinforces the need for the facilitation of



face-to-face communication, while the use of teams and teamwork provide an excellent mechanism for nurturing effective knowledge sharing. Convincing people to share their knowledge requires a combination of new processes, and a level of trust among individuals and managers. Inculcating trust is even more crucial while inducting new employees and individuals must feel comfortable that they will not lose their value to the company after sharing their knowledge with others and committing it to electronic form. This requires a 'no-blame' culture within the organisation and a rewards system that the top management needs to set out with a clear statement of values. The organisation's vision and value statements should clearly represent the commitment towards a knowledge culture, and be effectively communicated throughout the entire organisation.

Thus the research provides organisations with a new long-term perspective and better understanding of the interactions and relationship between software development, project management and knowledge management processes. Organisations need to develop capabilities to mobilise and utilise their existing knowledge resources and integrate them with new knowledge in a dynamic manner. Software project organisations need to adopt a continuous view of the development effort and manage context, provide feedback, facilitate interaction, and enable the flow of knowledge. This will lead to the creation, sharing and integration of tacit and explicit knowledge, and will ensure that the right knowledge is available to the right person at the right time during the software development effort. Since projects are temporal and constrained, adopting a long term perspective and providing the infrastructure for supporting and developing knowledge management processes, offers project-based organisations a path for continuous improvement.

Reich (2007) observes that 'one potential area through which projects could be understood and potentially improved is to conceptualise them as areas in which knowledge is generated and exploited and learning is essential for success. The foundations for this approach are found both in knowledge management and organisational learning literatures'. Reich (2007) further points out that the current literature mainly focuses on permanent organisations and fixed teams. The model presented and validated in this research offers the mechanism for facilitating such improvement within project management by focusing on all project teams through the adoption of a long term perspective, thus offering to harvest the benefits of learning across the organisation whilst also offering the potential to improve the profession and practice of project management.

## 11.5 Future Work

Several possibilities for future research emerge from the results of the current study. The interviews and observations during the case-study were conducted within a single organisation. The study did not attempt to isolate specific conditions that may tend to moderate the findings within a specific organisation. A focused study within several organisations, combined with an objective evaluation of the flow of knowledge and capability support within the various knowledge management initiatives, would provide useful follow-up research. Also, the model presented in this research has been proposed and validated for software development activities. Interesting research possibilities exist to extend and test the model within other developmental domains and industry. Therefore studies need to be conducted to look at organisations in other sectors to determine if the same practices apply.

Additionally, a research possibility is to examine how the different stages of KM capability development relate to an organisation's progress towards KM maturity, as suggested by Baskerville and Pries-Heje (1999) and Kulkarni and St Louis (2003). Both of the mentioned studies provide theoretical models that borrow from and match the CMM framework's five maturity levels. Further research is required to establish whether KM maturity is a staged progress along five levels, or does progress along the four states of KM capability to an organisational state achieve KM maturity.

There was no attempt to categorise the findings based on the size of the organisation. Opportunities for similar research appear to exist in this area, to determine if the research factors differ based on organisation size or structure. While this study was focused on the flow of knowledge and the KM capabilities that facilitate and support this flow within knowledge management initiatives, there is evidence in the literature that an effective knowledge management strategy may itself tend to enhance the flow of knowledge. Therefore, a longer-term study examining the changes in the flow of knowledge before and after implementing a knowledge management initiative would yield useful and interesting results.

Finally, further work is required to develop measures to determine the benefit of implementing a KM programme. As observed in the case study, organisations assume and expect a KM programme to provide benefits. However, further research is required to

determine and establish the impact a KM programme has on the work practices, benefits, and resources of an organisation.

## **11.6 Chapter Summary**

This chapter summarised the main findings and contribution of the thesis. This thesis validates feedback and interaction between the functional areas of the operations, which maybe structured implicitly, of a software development organisation. The thesis identifies the flow of knowledge within and between these functional areas, and the knowledge management infrastructure and process capabilities that facilitate and support this flow. The thesis further addresses the issues involved in the development of these capabilities. The chapter also detailed opportunities to further the work presented in this thesis. The findings of this research are based upon a case study conducted at a large software organisation. It is vital therefore that more work is undertaken to further this rigorous and imaginative research with regard to other possible initiative developments in organisations of different size and structure covering other sectors to determine the universal value of the proposition put forward in this work.

## II. References

Abdel-Hamid, T.K, and Madnick, S, 'Software Project Dynamics: An Integrated Approach,' Prentice-Hall Software Series, 1991.

Alavi, M., KPMG Peat Marwick U.S.: One Giant Brain, Harvard Business School, Case 9-397-108, 1997.

Alavi M, 'Managing Organizational Knowledge,' in Framing the Domains of IT Management, RW Zmud (ed), Cincinnati, OH: Pinnaflex Educational Resources, 2000.

Alavi, M, and Carlson, P, 'A Review of MIS Research and Disciplinary Development,' Journal of Management Information Systems, Vol 8, No. 4, pp 45-62, 1992.

Alavi, M and Leidner, D.E, 'Knowledge Management Systems: Issues, Challenges and Benefits,' Communications of the Association for Information Systems, Vol 1, Article 7 Feb 1999.

Alavi, M and Leidner, D.E, 'Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues,' MIS Quarterly Vol 25 No 1, pp 107-136, March 2001.

Alavi, M and Tiwana, A, 'Knowledge Management: The Information Technology Dimension,' In The Blackwell Handbook of Organizational Learning and Knowledge Management, Eds Easterby-Smith, M and Lyles, M.A, Blackwell Publishing, 2003.

American Productivity and Quality Centre (APQC), Knowledge Management: Consortium Benchmarking Study Final Report 1996; Available from <http://www.store.apqc.org/reports/Summary/know-mng.pdf>, accessed May 2004.

Aurum, A. 'Supporting Structures for Managing Software Engineering Knowledge.' In Aurum, A., Jeffery, R., Wohlin, C. and Handzic, M (Eds) Managing Software Engineering Knowledge, Springer-Verlag, 2003.

- Baker, W.E, 'The Network Organisation in Theory and Practice,' *Networks and Organisations: Structure, Form and Action*, Harvard Business School Press 1992.
- Barley, S, 'Technicians in the Workplace: Ethnographic Evidence for Bringing Work into Organisation Studies,' *Administrative Science Quarterly*, Vol 41, No 1, pp 146-162. 1996.
- Barney, J 'Firm Resources and Sustained Competitive Advantage,' *Journal of Management*, Vol 17 pp 99-120, 1991.
- Baskerville, R, and Pries-Heje, J, 'Knowledge Capability and Maturity in Software Management,' *The DATA BASE for Advances in Information Systems*, Vol 30, No 2, Spring 1999.
- Bassi, L, 'Harnessing the Power of Intellectual Capital,' J. Cortada and J. Woods Eds, *The Knowledge Management Yearbook 1999-2000*, Boston: Butterworth Heinmann: 422-431. 1999.
- Benbasat, I, Goldstein, D.K, and Mead, M, 'The Case Research Strategy in Studies of Information Systems,' *MIS Quarterly*, Vol 11, No 3. pp 369-86, 1987.
- Berger, P.L and Luckmann, T, *The Social Construction of Reality: A Treatise in the Sociology of Knowledge*. Garden City, NY, Doubleday, 1966.
- Berkun, S, 'The Art of Project Management,' O'Reilly Media, Inc, 2005.
- Blacker, F, Reed, M, and Whitaker, A, 'Editorial: Knowledge and the Theory of Organizations,' *Journal of Management Studies*, Vol 30, No 6, pp 851-861, 1993.
- Boehm, B, 'A Spiral Model of Software Development and Enhancement', *IEEE Computer*. Vol 21 No 5, pp 61-72, 1988.
- Borghoff, U.M, Pareschi, R, *Information technology for Knowledge Management*. Springer, Berlin Heidelberg New York, 1998.
- Bredillet, C.N, 'Projects are Producing the Knowledge which are Producing the Projects.....' *Proceedings of IPMA*, Budapest, 2004.

Brossler P, 'Knowledge Management at a Software Engineering Company – An Experience Report,' Proceedings of the Workshop on Learning Software Organisations. Kaiserslautern, Germany, pp 163-170, 1999.

Bryman, A, 'Research Methods and Organisation Studies,' Routledge, 1989.

Busch, P and Richards, D, 'Graphically Defining Articulate Tacit Knowledge,' Conferences in Research and Practice in Information Technology, Vol 2, pp 51-60, 2000.

Busch, P, Richards, D and Dampney, C.N.G, 'Visual Mapping of Articulate Tacit Knowledge,' Conferences in Research and Practice in Information Technology, Vol 9, pp 37- 47, 2001.

Castells, M, 'The Information Age: Economy, Society, and Culture. Volume 1: The Risk of the Network Society,' Blackwell, Oxford, 1996.

Ciborra, C.U, 'Groupware and Teamwork: Invisible Aid or Technical Hindrance?' Wiley, Chichester, 1996.

Cohen, D, 'Toward a Knowledge Context: Report on the First Annual UC Berkley forum on Knowledge and the Firm,' California Management Review, Vol 40, No 3, pp 22-39, 1998.

Cohen, W.M and Levinthal, D.A, 'Absorptive Capacity: A New Perspective on Learning and Innovation,' Administrative Science Quarterly, Vol 35, No 1, pp 128-152, 1990.

Concise Oxford English Dictionary, Oxford University Press, Oxford, 2002.

Constant, D, Sproull, L and Kiesler, S, 'The Kindness of Strangers: The Usefulness of Electronic Weak Ties for Technical Advice, Organisation Science, Vol 7, No 2, pp 199-135, 1996.

Cook, S and Brown, J.S, 'Bridging Epistemologies: The Generative Dance between Organizational Knowledge and Organizational Knowing,' Organization Science Vol 10 pp 381-400, 1999.

Cryer, P (2000), 'The Research Student's Guide to Success, Open University Press, McGraw-Hill Education, 2000.

Curtis, B, Krasner, H, and Iscoe, N, 'A Field Study for the Software Design Process for Large Systems,' Communications of the ACM, Vol 31, No 11, pp 68-87, 1988.

Dalcher, D, 'Towards Continuous Development: A Dynamic Process Perspective.' Information Systems Development: Advances in Methodologies, Components and Management, M. Kirikova, J. Grundspenkis, W. Wojtkowski, W. G. Wojtkowski, S. Wrycza, and J. Zupacic Eds, Kluwer Press, pp. 53-68, 2002a.

Dalcher, D, 'Life Cycle Design and Management,' Project Management Pathways: A Practitioner's Guide, M Stevens et al Eds, APM Press, High Wycombe, pp 60-1 – 60-30, September 2002b.

Dalcher, D, 'Maintaining Quality Beyond Development: Towards a Dynamic Perspective,' Proceedings 12th International Conference on Software Quality, 12ICSQ, Ottawa, Ontario, Canada, ASQ Press, October 2002c.

Dalcher, D, 'Project Management for Computing Students,' Middlesex University Press, 2003a.

Dalcher, D, 'Beyond Normal Failures: Dynamic Management of Software Projects,' Technology Analysis and Strategic Management, Vol 15, No 4, Dec 2003b.

Dalcher, D, 'Software Development for Dynamic Systems,' Developments in Metainformatics, LNCS, Springer Verlag, Peter J Numberg Ed, pp 58-75, June 2003c.

Dalcher, D, 'Dynamic Systems Development: Towards Continuity, Growth and Evolution'. 10th IEEE Engineering of Computer Based Systems Symposium (ECBS03), April 2003, Huntsville, Alabama, IEEE Press, Los Alamitos, pp 62-71, 2003e.

Dalcher, D, 'Design for Life: Invoking Continuity and Dealing with Change.' Journal of Applied Systems Studies, JASS Special issue on Living, Evolutionary and Tailorable Information Systems: Development Issues and advanced Applications, Vol 5, No 1, 2005a.

Dalcher, D, Design for Evolution: Fostering Continuity and Responding to Change". in Managing Corporate Information Systems Evolution and Maintenance. Khaled Khan and Yan Zhang Eds, Idea Publishing, pp 24-50, 2005b.

Dalcher, D, 'Maturity in IT Projects,' Project Management Days, PMT Age. Vienna, Austria, June 2005c.

Dalcher, D, and Brodie, L, 'Successful IT Projects,' Middlesex University Press, 2007.

Davenport, T.H and Prusak, L 'Working Knowledge,' Harvard Business School Press, Boston 1998.

Davis, R.C, 'The Fundamentals of Top Management,' New York: Harper and Brothers, 1951, (quoted in Cleland DI. Project Management: Strategic Design and Implementation, 3rd Eds. New York: McGraw-Hill, p-3), 1999.

Davis, T.R, 'The Influence of the Physical Environment in Offices,' Academy of Management Review, Vol 9, pp 271-283, 1984.

DeLong, D, 'Building the Knowledge Based Organization: how Culture Drives Knowledge Behaviours,' Working Paper, Ernst & Young's Centre for Business Innovation, Boston, 1997.

Deming, W.E, 'Out of the Crisis,' MIT Centre for Advanced Engineering Study, 1986.

Demsetz, H, 'The Theory of the Firm Revisited,' in The Nature of the Firm, J. Williamson and S. Winter Eds, Oxford University Press, New York, 1991.

Desouza, K.C, 'Facilitating Tacit Knowledge Exchange,' Communications of the ACM, Vol 46, No 6, pp 85 -88, 2003.

Dretske, F. 'Knowledge and the Flow of Information,' MIT Press, Cambridge, MA, 1981.

Dyba, T. A Dynamic Model of Software Engineering Knowledge Creation. Aurum. A., Jeffery, R., Wohlin, C. and Handzic, M (Eds) Managing Software Engineering Knowledge. Springer-Verlag, 2003.



- Earl, M, 'Knowledge Management Strategies: Toward a Taxonomy.' *Journal of Management Information Systems*, Vol 18. No 1, pp 215-233, Summer 2001.
- Easterby-Smith, M and Lyles, M.A, 'Introduction: Watersheds of Organisational Learning and Knowledge Management,' In *The Blackwell Handbook of Organizational Learning and Knowledge Management*, Eds Easterby-Smith, M and Lyles, M.A, Blackwell Publishing, 2003.
- Eisenhardt, K.M, 'Building Theories from Case Study Research.' *Academy of Management Review*, Vol 14. No 4, pp 532-50, 1989.
- Eskerod, P and Skriver, H.J, 'Organizational Culture Restraining In-house knowledge Transfer Between Project Managers-A Case Study,' *Project Management Journal*, vol. 38, no. 1, pp. 110-122, March 2007.
- Fayyad, U and Uthurusamy, R, 'Data Mining and Knowledge Discovery in Databases,' *Communications of the ACM*, Vol 39, No 11, pp 24-26, 1996.
- Fichman, R.G, 'The Diffusion and Assimilation of Information Technology Innovations,' Pinnaflex Educational Resources, Inc, [www2.bc.edu/~fichman/rob.html](http://www2.bc.edu/~fichman/rob.html), Accessed 20/05/03, 1999, accessed May 2003.
- Fong, P.S.W, Co-creation of Knowledge by Multidisciplinary Project Teams, in *Management of Knowledge in Project Environments*, Love, P.E.D, Fong, P.S.W. and Irani, Z Eds, Elsevier Butterworth-Heinemann, 2005
- George, J.F and Tyran, C.K, 'Expert Systems and Organizations: Predictions and Evidence,' *Accounting, Management and Information Technologies*, Vol 3, No 3, pp 173-189, 1993.
- Gilb, T, 'Principles of Software Engineering Management,' Addison Wesley, 1988.
- Glaser, B.G and Strauss, A, 'The Discovery of Grounded Theory: Strategies for Qualitative Research,' Aldine, Chicago, 1967.

Glass, R.L, Ramesh, V, and Vesey, I, 'An Analysis of Research in Computing Disciplines,' *Communications of the ACM*, Vol 47, No 6, pp 89-94. June 2004.

Gold, A.H, Malhotra, A, and Segars, A.H, 'Knowledge Management: An Organizational Capabilities Perspective,' *Journal of Management Information Systems*, Vol18, No 1, pp 185-214, Summer 2001.

Goldstein, M and Goldstein, I.F, 'How we Know: An Exploration of the Scientific Process,' Plenum Press, New York, 1978.

Gonclaves, M, 'Managing Virtual Projects,' McGraw-Hill, 2005.

Grant, R.M 'Prospering in Dynamically-Competitive Environments: Organizational Capability as Knowledge Integration,' *Organization Science* Vol7 No 4, pp 375-387, July-August 1996.

Grant, R.M 'Toward a Knowledge-based Theory of the Firm,' *Strategic Management Journal* Vol 17, Winter Special Issue, pp 109-122, 1996.

Grady, R.B, 'Successful Software Process Improvement,' Prentice-Hall, Upper Saddle River, NJ, 1997.

Gruenfeld, D.H, Mannix, E.A, Williams, K.Y, and Neale, M.A, 'Group Composition and Decision Making: How Member Familiarity and Information Distribution Affect Process and Performance,' *Organisational Behaviour and Human Decision Process*, Vol 67, pp 1-15, 1996.

Gupta, A and Govindarajan, V, 'Knowledge Flows within Multinational Corporations,' *Strategic Management Journal*, Vol 21, pp 473-496, 2000.

Hackbarth, G. 'The Impact of Organizational Memory on IT Systems,' *Proceedings of the Fourth Americas Conference on Information Systems*, E. Hoadley and I. Benbasat Eds pp 588-590, August 1998.

Hansen, M, 'The Search-Transfer Problem: The Role of Weak Ties in Sharing Knowledge Across Organizational Subunits,' *Administrative Science Quarterly*, Vol 44, pp 83-111, 1999.

Hansen, M, Nohira, N and Tierney, T, 'What is Your Strategy for managing Knowledge?' *Harvard Business Review*, pp 106-116, March –April 1999.

Hawryszkiewicz, I.T, 'Knowledge Sharing Through Workspace Networks,' *Communications of the ACM*, Vol 1, pp 79-85, 1999.

Henninger, S, 'Tools Supporting the Creation and Evolution of Software Development Knowledge,' 12<sup>th</sup> International Conference on Automated Software Engineering, Lake Tahoe, NV, pp 46-53, 1997.

Holmquist, M, *Experiential Learning Processes of Exploitation and Exploration Within and Between Organisations: An Empirical Study of Product Development*, *Organisation Science*, Vol 15, No 1, pp 70-81, 2004.

Holtham, C and Courtney, N, 'The Executive Learning Ladder: A Knowledge Creation Process Grounded in the Strategic Information Systems Domain,' in *Proceedings of the Fourth Americas Conference on Information Systems*, E. Hoadley and I. Benbasat Eds, Baltimore, MD, pp 594-597, 1998.

Hovarth, L, Callahan, J.L, Croswell, C and Mukri, G, 'Team Sense-making: An Imperative for Individual and Organisational Learning', In Holton, E.F (ed) *Proceedings of the Academy of Human Resource Development*, Minneapolis, MN, USA, pp 415-421, February-March, 1996.

Iansiti, M and MacCormack, A. 'Developing Products on Internet Time', *Harvard Business Review*, Vol 75, No 5, pp 108-117, 1997.

Inmon, W, 'The Data Warehouse and Data Mining,' *Communications of the ACM*, Vol 39, No 11, pp 49-50, 1996.

Ivers, J, 'Bringing Out Brilliance: Enabling Knowledge Creation in the Notes/Domino Environment,' *Enterprise Solutions*, pp 24-27, November/December 1998.

Jalote, P, CMM in Practice, Pearson Education Inc, 2000

Jalote, P. 'Knowledge Infrastructure for Project Management,' In Aurum, A., Jeffery, R., Wohlin, C. and Handzic, M, Eds Managing Software Engineering Knowledge, Springer-Verlag, 2003.

Johnson, R.A, 'Negotiation Basics,' Sage Publications, Inc, 1993.

Karsten, H, 'Collaboration and Collaborative Information technologies: A review of the Evidence,' The Data Base for Advances in Information Systems, Vol 30, No 2, pp 44-65, 1999.

Kast, F. E. and Rozenzweig, J. E. Organization and Management: A Systems and Contingency Approach, 4<sup>th</sup> edition McGraw-Hill, New York, 1985.

Kerth, N, Project Retrospectives,' Dorset House Publishing Co Inc, US. 2001.

Kerzner, H, Project Management, A Systems Approach to Planning, Scheduling and Controlling, John Wiley & Sons, 2003.

Kerzner, H, Advanced Project Management, Best Practices on Implementation, John Wiley & Sons, 2004.

Khalifa, M, and Liu, V, Determinants of Successful Knowledge Management Programs, Electronic Journal on Knowledge Management, Vol 1, No 2, pp 103-112, 2003.

King, W.R, Marks. Jr, P.V, and McCoy, S, 'The Most Important Issues in Knowledge Management,' Communications of the ACM, Vol 45, No 9, pp 93-97, 2002.

Kirk, J and Miller, M.L, 'Reliability and Validity in Qualitative Research,' Sage, 1986.

Kock, N, and McQueen, R, 'Knowledge and Information Communication within Organisations: An Analysis of Core, Support and Improvement Process,' Knowledge and Process Management, Vol 5, No 1, pp 29-40, 1998.

Kock, N, and McQueen, R.J, 'Groupware Support as a Moderator of Interdepartmental Knowledge Communication in Process Improvement Groups: An Action Research Study.' *Information Systems Journal*, Vol 8, pp 183-198, 1998.

Kolb, D, 'Experiential Learning: Experience as a Source of Learning and Development.' Engelwood Cliffs, NJ: Prentice-Hall, 1984.

Koskinen, K.U, 'Evaluation of Tacit Knowledge Utilisation in Work Units,' *Journal of Knowledge Management*, Vol 7, No 5, pp 67-81, 2003.

Koskinen, K.U, 'Knowledge Management to Improve Project Communication and Implementation,' *Project Management Journal*, Vol 35, No 1, pp 13-19, 2004.

Kulkarni, U, and St Louis, R, 'Organizational Self Assessment and Knowledge Management Maturity,' *Proceedings Americas Conference on Information Systems*, August 2003.

Kvale, S, 'Interviews: An Introduction to Qualitative Research Interviewing,' Sage Publications, Thousand Oaks, CA, 1996.

Lado, A. and Zhang, M, 'Expert Systems, Knowledge Development and Utilization, and Sustained Competitive Advantage: A Resource-Based Model,' *Journal of Management*, Vol 24, No 4, pp 489-509, 1998.

Leonard-Barton, D, 'Wellsprings of Knowledge: Building and Sustaining the Source of Innovation,' Harvard Business School Press, Boston, 1995.

Leonard, D and Sensiper, S, 'The Role of Tacit Knowledge in Group Innovation,' *California Management Review*, Vol 40, No 3, pp 112-132, 1998.

Leseure, M.J and Brookes, N.J, 'Knowledge Management Benchmarks for Project Management,' *Journal of Knowledge Management*, Vol 8, No 1, pp 103-116, 2004.

Levitt, B and March, J.G, *Organisational Learning*, *Annual Rev. Sociology*, Vol 14. pp 319-340, 1988.

- Lewis, J.P, 'Fundamentals of Project Management,' AMACOM. 2002.
- Li, M and Gao, F, 'Why Nonaka Highlights Tacit Knowledge: A Critical Review,' Journal Of Knowledge Management, Vol 7, No 4, pp 6-14, 2003.
- Liebowitz, J. and Wilcox, L, 'Knowledge management and its Integrative Elements,' Boca Raton: CRC Press, 1997.
- Lindvall, M and Rus, I, 'Knowledge Management for Software Organisations,' In Aurum, A., Jeffery, R., Wohlin, C. and Handzic, M, Eds Managing Software Engineering Knowledge, Springer-Verlag, 2003.
- Loeb, K.A, Rai, A, Ramaprasad, A and Sharma, S, 'Design, Development and Implementation of a Global Information Warehouse: A Case Study at IBM,' Information Systems Journal, Vol 8, No 4, pp 291-311, 1998.
- Lofland, J and Lofland, L, 'Analysing Social Settings,' Belmont, CA. Wadsworth, 1984.
- Lofland, J, 'Analytic Ethnography: Features, Failings, and Futures,' Journal of Contemporary Ethnography, Vol 21, No 1, pp 30-67. 1995.
- Love, P.E.D, Smith, J and Li, H, 'The Propagation of Rework Benchmark Metrics for Construction,' International Journal of Quality and Reliability Management, Vol 16, No 7, pp 638-658, 1999.
- Lynne Markus, M, 'Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success,' Journal of Management Information Systems, Vol 18, No 1, pp 57-93, Summer, 2001.
- Machlup, F 'Knowledge: Its Creation, Distribution and Economic Significance, Vol 1' Princeton University Press, Princeton, NJ 1983.
- Malhotra, Y, 1997 <http://www.brint.com/interview/maeil.htm>, accessed on 01/08/03
- March, J.G, The Pursuit of Organisational Intelligence. Blackwell Business, Oxford, UK. 1999.

Mao, J.Y and Benbasatm I, 'Contextualized Access to Knowledge: Theoretical Perspectives and a Process-Tracing Study,' *Information Systems Journal*. No 8, pp 217-239, 1998.

McNurlin, B. and Sprague, R, 'Information Systems Management in Practice 5/E,' Upper Saddle River, NJ: Prentice Hall, 2001.

Meredith, J.R, and Mantel, J.S, 'Project Management: A Managerial Approach,' Wiley, New York, 1995.

Miles, M. B, and Huberman, A. M, 'Qualitative Data Analysis. An Expanded Sourcebook,' Sage Publications, London, 1994.

Mitchell, T. R. and Larson J. R., *People in Organization: An Introduction to Organizational Behaviour*, 3<sup>rd</sup> edition McGraw-Hill, New York, 1987.

Morgan, G. 'Images of Organisations,' Sage Publications, Inc, 1997.

Morrison, R and Kennedy, J Eds 'Advances in Databases: Proceedings of the 14<sup>th</sup> British National Conference on Databases', BNCOD 14 Edinburgh UK, New York: Springer, 1996.

Myers, W, 'MCC: Planning the Revolution in Software,' *IEEE Software*, November. 1985.

Nonaka, I, 'A Dynamic Theory of Organisation Knowledge Creation,' *Organisation Science*, Vol 5, No 1, pp 14-37, 1994.

Nonaka, I and Takeuchi, H 'The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation,' Oxford University Press, New York, 1995.

Nonaka, I. and Konno, N. 'The Concept of 'Ba', Building a Foundation for Knowledge Creation,' *California Management Review*, Vol 40, No 3, pp 40-54, 1998.

Nonaka, I and Nishiguchi, T. 'Social, Technical, and Evolutionary Dimensions of Knowledge Creation,' In I. Nonaka and T. Nishiguchi Eds, *Knowledge Emergence: Social,*

Technical, and Evolutionary Dimensions of Knowledge Creation. New York, Oxford University Press, pp 286-289, 2001.

Oates, B. J, 'Researching Information Systems and Computing,' Sage, London, 2006.

O'Dell, C. and Grayson, C. "If only we knew what we know: Identification and transfer of internal best practices," California Management Review Vol. 40, No. 3, pp. 154-174, 1998.

O'Leary, D, 'Using AI in Knowledge Management: Knowledge Bases and Ontologies,' IEEE Intelligent Systems Vol 13 pp 34-39 1998.

Orr, J.E, 'Sharing Knowledge, Celebrating Identity: Community Memory in a Service Culture,' in D. Middleton and D. Edwards Eds, Collective Remembering, Newbury Park, CA: Sage, 1990.

Pentland, B. T, 'Information Systems and Organizational Learning: The Social Epistemology of Organizational Knowledge Systems,' Accounting, Management and Information Technology, Vol 5, No 1, pp 1-21, 1995.

Perry, D.E, Sim, S.E and Easterbrook, S, 'Case Studies for Software Engineers,' International Conference on Software Engineering (ICSE), Edinburgh, Scotland, May 2004. Viewed 25/11/2005 at <http://www.ece.utexas.edu/~perry/work/papers.html>

Peters, T, 'Get Innovative or Get Dead,' California Management Review, Vol 33, No 1, pp 9-26, 1990.

Pfeffer, J, and Sutton, R.I, 'The Knowledge-doing Gap: How Smart Companies Turn Knowledge into Action,' Harvard Business School Press, Boston, 2000.

Piaget J, 'Genetic Epistemology,' Columbia University Press New York, USA 1970.

Polanyi, M, 'The Tacit Dimension,' Routledge and Keon Paul, London 1967.

Politis, J.D, 'The Connection between Trust and Knowledge Management: What are its Implications for Team Performance,' Journal of Knowledge Management, Vol 7, No 5, pp 55-66, 2003.



- Popper, K.R, 'The Logic of Scientific Discovery,' Hutchinson & Co (Publishers) Ltd. London, 1959.
- Pressman, R.S, 'Software Engineering – A Practitioner's Approach,' The McGraw-Hill Companies, Inc, 1997.
- Reich, B. L, 'Managing Knowledge and Learning in IT Projects: A Conceptual Framework and Guideline for Practice', Project Management Journal, Vol 38, No. 2. pp. 5-17. June 2007.
- Richardson, G.P, and Pugh, G.L. III, 'Introduction to Systems Dynamics Modelling and DYNAMO,' Cambridge, MA, MIT Press, 1981.
- Richman, L, 'Improving Your Project Management Skills,' AMACOM, 2006.
- Royce, W.W, 'Managing the Development of Large Software Systems,' IEEE WESCON, pg 1-9, August 1970.
- Rubenstein-Montano, R., Liebowitz, J., Buchwalter, J. and McGraw, D, 'A Systems Thinking Framework for Knowledge Management,' Available: <http://userpages.umbc.edu/~buchwalt/papers/papers.html>, 2000, accessed July 2003.
- Rubinstein, M.F, and Pfeiffer, K, Concepts in Problem Solving, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- Senge, P, 'The Fifth Discipline: The Art and Practice of the Learning Organization,' New York: Doubleday/Currency 1990.
- Shewhart, W. A, 'Statistical Method from the Viewpoint of Quality Control,' Dover, New York, 1939.
- Shukla, A, and Srinivasan, R, 'Designing Knowledge Management Architecture,' Sage, California, 2002.
- Simon, H.A, 'The New Science of Management Decision,' Prentice-Hall Inc. 1977.

- Skyrme, D, and Amidon, D, 'New Measures of Success,' *Journal of Business Strategy*, pp 20-24, January/February 1998.
- Sommerville, I, 'Software Engineering,' Pearson Education, UK, 2003.
- Spender, J.C 'Making Knowledge the Basis of a Dynamic Theory of the Firm,' *Strategic Management Journal* Vol 17, pp 45-62, Special Issue 1996.
- Spender, J.C 'Organizational Knowledge, Learning and Memory: Three Concepts in Search of Theory,' *Journal of Organizational Change Management* Vol 9, pp 63-78, 1996.
- Spradley, J.P 'The Ethnographic Interview' New York; Holt, Rinehart and Winston, 1979.
- Stear, E, and Bair, J, 'Information Management is not Knowledge Management,' Note No. TU-03-2524, Gartner Group 17 December 1997, Available [www.gartner.com](http://www.gartner.com)
- Stein, E.W and Zwass, V, 'Actualizing Organizational Memory with Information Systems,' *Information Systems Research*, Vol 6, No 2, pp 85-117, 1995.
- Sternberg, R, Wagner, R, Williams, W and Horvath, J, 'Testing Common Sense,' *American Psychologist*, Vol 50, No 11, pp 912 – 927, 1995.
- Stewart, T. A, *Intellectual Capital*, Doubleday division of Random House, Inc, 1997.
- Styhre, A, 'Knowledge Management beyond codification: knowing as practice/concept,' *Journal of Knowledge Management*, Vol 7 No 5, pp 32-40, 2003.
- Sunassee, N.N and Sewry, D.A, 'A Theoretical Framework for Knowledge Management Implementation,' *Proceedings of SAICSIT*, pp235-245, 2002.
- Taylor, S.J and Bogdan, R, 'Introduction to Qualitative Research Methods,' John Wiley and Sons, Inc, 1998.
- Teece, D, 'Capturing Value from Knowledge Assets: the New Economy. Markets for Know-How and Intangible Assets,' *California Management Review*, Vol 40, No 3, pp 55-79, 1998.

- Tiwana, A, 'The Knowledge Management Toolkit: Practical Techniques for Building a Knowledge Management System,' Upper Saddle River, NJ: Prentice-Hall. 1999.
- Tsoukas, H, 'Where Does New Organisational Knowledge Come From?' Keynote address at the International Conference Managing Knowledge: Conversations and Critiques. Leicester University, April 10-11, 2001.
- Vance, D.M 'Information, Knowledge and Wisdom: The Epistemic Hierarchy and Computer-Based Information System,' Proceedings of the Third Americas Conference on Information Systems, B. Perkins and I. Vessey Eds, Indianapolis, IN, August 1997.
- Van der Spek, R and Spijkervet, A, 'Knowledge Management: Dealing Intelligently with Knowledge,' J. Liebowitz and L. Wilcox Eds, Knowledge Management and its Integrative Elements, Boca Raton: CRC Press: 31-59 1997.
- Vera, D and Crossan, M, 'Organisational Learning and Knowledge Management: Toward an Integrative Framework,' In The Blackwell Handbook of Organizational Learning and Knowledge Management, Eds Easterby-Smith, M and Lyles, M.A, Blackwell Publishing, 2003.
- Von Krogh, G and Roos, J, 'Five Claims of Knowing,' European Management Journal, Vol 14, pp 423-426, 1996.
- Von Grogh, G. 'Care in Knowledge Creation,' California Management Review Vol 40 No 3 pp 133-153, 1998.
- Von Krogh G, Ichijo, K and Nonaka, I., 'Enabling Knowledge Creation,' Oxford University Press, 2000.
- Walsh, J.P and Ungson, G.R, 'Organizational Memory,' Academy of Management Review, Vol 16, No 1, pp 57-91, 1991.
- Walsham, G, 'Knowledge Management: The Benefits and Limitations of Computer Systems,' European Management Journal, Vol 19, No 6, pp 599-608, 2001.

Walz, D.B, Elam, J.J and Curtis, B, 'Inside a Software Design Team: Knowledge Acquisition, Sharing and Integration,' Communications of the ACM, Vol 36, No 10. pp 63-77, 1993.

Wasko, M and Faraj, S, "It's What One Does": Why People Participate and Help Others in Electronic Communities of Practice, Journal of Strategic Information Systems. Vol 9, No 2-3, pp 155-173, 2000.

Wenger, E, and Snyder, W, 'Communities of Practice: The Organizational Frontier,' Harvard Business Review, Vol 78, No 1, pp 139-145, 2000.

Wiig, K, Knowledge Management: Foundations, Schema Press, Arlington, Texas, USA, 1993.

Wiig, K, 1996 'On the Management of Knowledge,' Available: [http://www.km-forum.org/what\\_is.htm](http://www.km-forum.org/what_is.htm), accessed on 05/08/03, 1996.

Yin, R.K, 'Case Study Research: Design and Methods,' Sage Publications, Inc, 2003.

Yin, R.K, 'Applications of Case Study Research', Sage Publications, Inc, 2003.

Zack, M.H, 'Developing a Knowledge Strategy,' California Management Review. Vol 41, No 3, pp 125-146, 1999.

Zack, M.H, 'Managing Codified Knowledge,' Sloan Management Review. pp 45-58, Summer 1999.

## Appendix 1 Knowledge Management Frameworks

Framework	Description
American Management Systems	(1) Find [create knowledge centres], (2) Organize [motivate and recognize people] and (3) Share
Arthur Andersen Consulting	(1) Evaluate, (2) Define the role of knowledge, (3) Create a knowledge strategy linked to business objectives, (4) Identify processes, cultures and technologies needed for implementation of a knowledge strategy and (5) Implement feedback mechanisms
Andersen Consulting	(1) Acquire, (2) Create, (3) Synthesize, (4) Share, (5) Use to Achieve Organizational Goals, (6) Environment Conducive to Knowledge Sharing
Dataware Technologies, Inc.	(1) Identify the Business Problem, (2) Prepare for Change, (3) Create the KM Team, (4) Perform the Knowledge Audit and Analysis, (5) Define the Key Features of the Solution, (6) Implement the Building Blocks for KM and (7) Link Knowledge to People
Buckley & Carter Centre for International Business, University of Leeds	Business process approach to knowledge management (no formal methodology but key knowledge processes are identified): (1) Knowledge Characteristics, (2) Value Added from Knowledge Combination, (3) Participants, (4) Knowledge Transfer Methods, (5) Governance and (6) Performance
The Delphi Group [18]	(1) Key Concepts and Frameworks for Knowledge Management, (2) How to Use Knowledge Management as a Competitive Tool, (3) The Cultural and Organizational Aspects of Knowledge Management, (4) Best Practices in Knowledge Management, (5) The Technology of Knowledge Management, (6)  Market Analysis, (7) Justifying Knowledge Management and (8)  Implementing Knowledge Management
Ernst and Young	(1) Knowledge Generation, (2) Knowledge Representation, (3) Knowledge Codification and (4) Knowledge Application
Holsapple and Joshi Kentucky Initiative for Knowledge Management	(1) Acquiring Knowledge [including Extracting, Interpreting and Transferring], (2) Selecting Knowledge [including Locating, Retrieving and Transferring], (3) Internalizing Knowledge [including Assessing, Targeting and Depositing], (4) Using Knowledge, (5) Generating Knowledge [including Monitoring, Evaluating, Producing and Transferring] and (6) Externalizing Knowledge [including Targeting, Producing and Transferring]
Holsapple and Joshi	(1) Managerial Influences [including Leadership,

Framework	Description
	Coordination, Control, Measurement], (2) Resource Influences [including Human, Knowledge, Financial, Material], (3) Environmental Influences [including Fashion, Markets, Competitors, Technology, Time, Climate] (4) Activities [including Acquire, Select, Internalize, Use], (5) Learning and Projection as Outcomes
Knowledge Associates	(1) Acquire, (2) Develop, (3) Retain and (4) Share
The Knowledge Research Institute Inc.	(1) Leverage Existing Knowledge, (2) Create New Knowledge, (3) Capture and Store Knowledge, (4) Organize and Transform Knowledge and (5) Deploy Knowledge.
Liebowitz	(1) Transform Information into Knowledge, (2) Identify and Verify Knowledge, (3) Capture and Secure Knowledge, (4) Organize Knowledge, (5) Retrieve and Apply Knowledge, (6) Combine Knowledge, (7) Learn Knowledge, (8) Create Knowledge [ loop back to (3) and (9) Distribute/Sell Knowledge
Liebowitz and Beckman	(1) Identify [Determine core competencies, sourcing strategy and knowledge domains], (2) Capture [Formalize existing knowledge], (3) Select [Assess knowledge relevance, value, and accuracy and resolve conflicting knowledge], (4) Store [Represent corporate memory in knowledge repository] (5) Share [Distribute knowledge automatically to users based on interest and work and collaborate on knowledge work through virtual teams], (6) Apply [Retrieve and use knowledge in making decisions, solving problems, automating or supporting work, job aids and training], (7) Create [Discover new knowledge through research, experimenting, and creative thinking] and (8) Sell [Develop and market new knowledge-based products and services]
Marquardt	(1) Acquisition, (2) Creation, (3) Transfer and Utilization and (4) Storage
Monsanto Company	No formal knowledge management methodology: Use learning maps, values maps, information maps, knowledge maps, measurements, and information technology maps.
The Mutual Group	Capital framework: (1) Gather Information [building an explicit knowledge infrastructure], (2) Learn [tacit knowledge development], (3) Transfer and (4) Act [developing capability through values deployment]
The National Technical University of Athens, Greece	(1) Context [generating knowledge], (2) Knowledge Management Goals [organizing knowledge], (3) Strategy [developing and distributing knowledge] and (4) Culture
O'Dell American Productivity and Quality Centre	(1) Identify, (2) Collect, (3) Adapt, (4) Organize, (5) Apply, (6) Share and (7) Create

Framework	Description
PriceWaterhouseCoopers	(1) Find, (2) Filter [for relevance], (3) Format [to problem]. (4) Forward [to right people] and (5) Feedback [from users]
Ruggles	(1) Generation [including Creation, Acquisition, Synthesis, Fusion, Adaptation], (2) Codification [including Capture and Representation] and (3) Transfer
Skandia	Universal Networking Intellectual Capital: Emphasizes (1) networking and knowledge sharing, (2) knowledge navigation by project teams, (3) intellectual capital development tool box
Van der Spek and de Hoog	(1) Conceptualize [including Make an inventory of existing knowledge and Analyze strong and weak points], (2) Reflect [including Decide on required improvements and Make plans to improve process], (3) Act [including Secure knowledge, Combine knowledge, Distribute knowledge and Develop knowledge] and (4) Review [including Compare old and new situation and Evaluate achieved results]
Van der Spek and Spijkervet	(1) Developing New Knowledge, (2) Securing New and Existing Knowledge, (3) Distributing Knowledge and (4) Combining Available Knowledge
Van Heijst et al. CIBIT, Netherlands	(1) Development [creating new ideas, analyzing failures and examining current experiences], (2) Consolidation [storing individual knowledge, evaluation and indexing], (3) Distribution [informing users] and (4) Combination [combining disparate information and increasing access to distributed data]
Wielinga et al. University of Amsterdam	Apply Common KADS methodology to knowledge management: (1) Conceptualize [identify/inventory, represent, classify], (2) Reflect [models of knowledge development and creation, models for identifying knowledge resources and results] and (3) Act [combine and consolidate knowledge, integrate knowledge, develop and distribute knowledge]
Wiig	(1) Creation and Sourcing (2) Compilation and Transformation, (3) Dissemination Application and (4) Value Realization

## Appendix 2 Project teams, community of practice, and Ba.

<b>Project Team</b>	<b>Community of Practice</b>	<b>Ba</b>
Members practice their jobs and learn by participating to the project team	Members learn by participating to the community and practicing their jobs	Members learn by participating to the <i>Ba</i> and practicing their jobs
Place where knowledge is created, where members learn knowledge that is embedded, and where knowledge is utilized	Place where members learn knowledge that is embedded in the community	Place where knowledge is created
Need of energy (forming the team) and then learning occurs	Learning occurs in any community of practice	Need of energy in order to become active
Boundary is set by the task and the project.	Boundary is firmly set by the task, culture, and history of the community	Boundary is set by its participants and can be changed easily. Here-and-now. Created, function, disappear
Membership fixed for the project duration (temporary nature). May vary depending the phases of the project.	Membership rather stable. New members need time to learn and fully participate.	Membership not fixed. Participants come and go.
Participants may relate/belong to the project team for the duration of the project but may belong/relate to the operational/functional organization (Department, contractors, suppliers...).	Participants belong to the community.	Participants relate to the <i>Ba</i> .

After Bredillet (2004)



## **Appendix 3      Refereed and Published Proceedings**

3.1. Dalcher, D and Sandhawalia, B, "The Dynamic Perspective: Incorporating Knowledge and Learning into the Software Development Process", Proceedings Software Quality Management (SQM) XII -- New Approaches to Software Quality, British Computer Society, Canterbury, UK, March-April 2004.

Software development projects present good opportunities for organisational learning and knowledge creation. However, modern software development projects are complex, and present a high level of uncertainty and ambiguity. These projects require a dynamic model that supports learning and knowledge creation, while working in an ill-structured and fast changing environment. This paper examines the capture and re-use of learning and knowledge by the Dynamic Feedback Model, which adopts a long-term perspective of software development.

3.2. Dalcher, D and Sandhawalia, B, "Learning and Knowledge Capture in Projects", Proceedings 5th European Conference on Knowledge Management (ECKM), CNAM, Paris, France, September-October, 2004.

Organisations need learning and knowledge to achieve their objectives; so do projects. Projects, by definition, are focused on change. They function in an uncertain and ambiguous environment, and need to be flexible and adaptive. Since projects are temporally limited, the opportunity to capture learning and knowledge is also limited. Projects, therefore, require different mechanisms for knowledge creation and sharing. This paper addresses the specific needs of projects and proposes a framework for the long-term management of knowledge in projects.

3.3. Sandhawalia, B and Dalcher, D, "Negotiation and Decision-Making within Software Projects", Proceedings Group Decision and Negotiation (GDN) Annual Conference, Vienna, Austria, July 2005.

Negotiation and decision-making are central to implementing software projects. The use of knowledge results in more effective negotiation and decision-making. This paper analyses

the creation and capture of knowledge within software development projects and discusses the central role of decision making in the development process, and how the effective use of knowledge helps improve decision-making. The paper views how the knowledge and decisions implemented can be provided greater visibility within the projects and communicated effectively.

3.4. Sandhawalia, B and Dalcher, D, "Knowledge Management Support for Software Project Organisations", Proceedings 4th Annual Conference on Information Science, Technology and Management (CISTM), Chandigarh, India, July 2006.

Software project organisations require knowledge to implement projects effectively. Software projects are organised around teams assembled specifically for the limited duration of the project and the development process relies upon knowledge and the creativity of the team members. This paper discusses the need to manage knowledge in software projects. The paper further analyses the creation of knowledge within software development projects, and views how this knowledge can be provided greater visibility and communicated effectively across the organisation. To achieve this the paper presents a framework that facilitates interaction and feedback in the developmental process, and enables effective knowledge management support for software project organisations.

3.5. Sandhawalia, B and Dalcher, D, "Knowledge and Decision-Making within Software Projects", Proceedings 15th International Conference on Information Systems Development (ISD), Budapest, Hungary, August-September, 2006.

The effective use of knowledge results in better decision-making within projects. This paper analyses the creation and capture of knowledge within software development projects and discusses the central role of decision making in the development process, and how the effective use of knowledge helps improve decision-making. The paper views in detail how decisions made within a software project can be leveraged effectively, and to achieve this, presents a framework that facilitates the central role of decision-making within the software development process.

3.6. Sandhawalia, B and Dalcher, D, "Knowledge Support for Software Projects", Proceedings 18th Annual Information Resources Management Association (IRMA) Conference, Vancouver, Canada, May 2007.

The unpredictable nature of software projects and the need for effective communication within project teams requires a framework for social interaction and feedback that results in better decision-making. This paper analyses the creation and capture of knowledge within software development projects and discusses the central role of decision making in the development process. The paper views how the knowledge generated within a software project can be provided greater visibility and communicated effectively, and to achieve this, presents a framework to facilitate social interaction and feedback during the development process.

3.7. Sandhawalia, B and Dalcher, D, "Knowledge Management Capability Framework", Accepted paper at Knowledge Management In Action (KMIA) Conference to be held in Milan, Italy, September 7-10, 2008.

This paper presents a Knowledge Management Capability framework based upon an empirical case study conducted at a CMMI Level 5 software project organisation. The paper discusses the development of the organisation's knowledge management (KM) initiative from its initial state, to an organisational state where the KM practices are institutionalised and embedded within the daily activities and work methods of the organisation. The organisation's KM initiative is analysed through the development of two KM capabilities, namely infrastructure and processes, which were examined in depth while conducting the case study, and are the basis for the KM Capability Framework. The resulting framework helps organisations to analyse any imbalance that may exist in their KM initiative and needs to be addressed. In doing so, the framework benefits organisations to make corrections and restore balance between KM infrastructure and process capability, thereby smoothening the path of successful KM implementation towards a state of organisational KM capability.

## Appendix 4      Sample Interviews

Interview with P B. 15 Jan 2006.

What is XYZ's knowledge strategy?

XYZ's knowledge strategy has evolved through knowledge usage over many years. XYZ has developed a knowledge culture at the local, branch, nodal and corporate levels. In a globally distributed organisational environment, each local centre has a knowledge champion who is well versed in the local culture. Local knowledge champions interact with branch knowledge managers, who in turn interact at the nodal and corporate levels leading to an overall knowledge owner. The overall corporate knowledge is managed and disseminated by the SEPG. The intranet portal is the primary tool for collating and disseminating corporate knowledge.

Since XYZ has employed a majority of its personnel after implementing the knowledge strategy (50% of the employees have been at XYZ for less than 3 years), the current knowledge culture is an integral part of their induction and training programs. Also, XYZ hires 'bench strength' that provides a bigger talent pool, ongoing training, and possibilities of job-rotation.

Implication: XYZ's knowledge strategy is hybrid, that is, it draws upon both the personalisation and codification strategies.

What obstacles or issues were faced while implementing the knowledge strategy?

Only technical issues were faced while implementing the knowledge initiative, *if any*. Training programs and a reward system were implemented, and goals set with regard to knowledge contribution. Personnel were educated on the features and benefits of the knowledge initiatives through innovative publicity such as printed mouse-pads.

How does knowledge flow within XYZ? Also, what are the requirements for a smooth flow of such knowledge?

Knowledge flows in XYZ as a part of the daily activities of XYZ employees. It flows in all directions. There is no model flow of XYZ knowledge as it is a vibrant organisation and knowledge flows across hierarchies and departments in a quasi structure.

How do interdepartmental teams communicate and coordinate within XYZ?

The primary mode of communication and coordination is the intranet portal.

How has the knowledge initiative (read intranet portal) modified XYZ' work methods and improved performance?

“ABC-portal is XYZ”.

“ABC-portal is the lifeline of XYZ.”

The intranet portal provides 24 hour access to resources that employees at XYZ would require.

ABC-portal helps with process changes, defect prevention, and technology changes. ABC-portal also helps in process, estimation and white paper differentiation.

How is the knowledge strategy reviewed?

New features are constantly added to ABC-portal. DEF, a new KMS has been added which is a central repository and a value addition to the existing search and retrieval features. Basically, it is a new structured KMS that consolidates all scattered assets into one system that caters to global needs of 85000 diverse employees. Assets are published in the KMS through review, and Meta-tags are mandatory.

How does XYZ as an organisation:

- learn
- identify patterns
- formalise routines

---

Interview Feb 16, 2006.

PB

What was the primary vision for the knowledge initiative at XYZ?

XYZ had many previous systems. Initially there was the paper-based system which involved established processes of submitting project reports, information and empirical data. These systems eventually became islands within the office automation systems. In 1992-93 XYZ had adopted the ISO 9000 model and most reports were in a fairly established format. These reports formed the basis of XYZ's asset library. I would therefore say that the XYZ's knowledge initiative evolved rather than starting of a recent knowledge vision. Also XYZ is currently into its fourth generation system and our *biggest challenge is scalability* of knowledge....from the initial knowledge requirements of 10000 employees, the system now has to cater to the knowledge needs of more than 80000 employees. Therefore XYZ now requires and needs a system that is *independent of people*, and that is what our latest knowledge effort DEFL has tried to address.

How would it be possible to facilitate tacit knowledge in a system independent of people?

Our latest KMS contains Project Profiles of all projects implemented by XYZ. These profiles contain details of the key contact person with regard to that particular project, and contact information of all individuals involved with the project, including clients.

So, even though you consider this system to be independent of people, it still is a system about people? In other words, it is a system that provides information about people so that individuals are able to contact each other?

Yes.

So you would agree that the system you have in place facilitates communication amongst people, which therefore helps in the transfer of tacit knowledge?

Yes. But also, I must mention here that the system carries information that is not more than 10 years old, as that information/knowledge would not be relevant in today's world regarding how we implement our projects. The project profiles are updated regularly especially in modules like QMS (Quality Management System).

How did XYZ develop an organisational knowledge culture? My question regarding organisational culture relates to issues such as Leadership, organisational structure, time allocation, Business process, office plan and mentoring.

XYZ organisational culture could be said to be all of three – informal, quasi-formal and formal. Knowledge flows in all directions at XYZ, across all levels through formal and informal channels.

Our organisational structure now has a CKO (Chief Knowledge Officer) who is the owner of the knowledge processes with Executive Sponsors, thereby reflecting the top management support and commitment to a knowledge culture. Knowledge Champions report to the CKO and there is a whole hierarchy of people who ensure that the knowledge flows in a process driven manner through the organisation. The sheer size and scale of our knowledge structure requires that a formal structure is in place to ensure that the knowledge needs are met, because relying only upon informal channels would mean that the knowledge exchange is ad-hoc. As you would have noticed our office plan is an open layout with a lot of places made available for people to have frequent meetings, both in a formal and informal setting.

An important aspect of (informal and formal) knowledge exchange at XYZ is individual visits and interactions. For example, when a Delhi based employee visits the Sydney or London office of XYZ, he/she interacts with XYZ employees at that location: again both formally and informally. XYZ views these visits as central to their knowledge exchange requirements. At a more formal level, local knowledge champions travel to formal knowledge (exchange) meetings which range from the zonal, regional, national and corporate levels.

Other formal channels of knowledge exchange at XYZ are Communities of Practice and Centres of Excellence. Such communities can be local or virtual and can be created within the portal, for example Architects Forum.

What issues/obstacles did XYZ face while implementing their knowledge initiative?

The major challenge was the sheer *SCALE* of the initiative – we had to cater to the knowledge needs of so many existing employees. We had to ensure that they can rely upon having the right knowledge available to them at the right time. New employees now joining the organisation are made familiar of our knowledge practices as a part of their induction program. But existing employees had to be made aware of the KMS and encouraged to use it. It was almost as if *we had to market the KMS to our employees*.

How did you address that (making existing employees aware of the KMS)?

We conducted seminars and training programs. Also we had publicity material made like posters and mouse pads. We also carried articles and prominent displays in our organisations monthly magazine.

What are the important features of XYZ's knowledge initiative?

The most important aspect of our knowledge initiative is our intranet portal – it is the lifeline of our knowledge needs. In addition to that we conduct technical seminars, and weekly meetings, and also have learning weeks and library weeks for our employees.

How has XYZ managed to measure the improvements brought about by implementing the knowledge initiative?

That is our next challenge and focus – to develop measures regarding the impact and benefits of implementing DEF. Currently the greatest indicator of the benefits of implementing DEF are CSI (Customer Satisfaction Index). We feel that we are delivering better quality software quicker to our clients. We also have weekly meetings to assess how the system is benefiting the organisation.

Yes, but you still do not have any empirical data on the time-saved in terms of software development, learning curve and quality improvement (lesser number of bugs etc)?

No.

Who at XYZ is able to understand the impact of changes brought about by implementing the knowledge initiative?

The most obvious impact would be the return on investment.

Yes, but as mentioned before, the knowledge initiative at XYZ evolved over the years therefore it would be difficult to reach an exact figure in terms of investment. Also, there are no measures in place to analyse the benefit in terms of the systems implementation. We will be developing parameters and criteria to measure the impact of implementing our KMS. An impact of our knowledge initiative has been the creation of the role of CKO, and one of our expectations from the CKO is the development of such parameters and criteria to analyse the knowledge benefits.



However, there is a change in the work methods of the organisation's business processes. The benefits that accrue are from the business perspective – implementing the knowledge initiative will/has definitely bring down the costs of developing software; knowledge and artefacts are reused and time is not spent validating them. Therefore lesser effort is required in developing software and lesser artefacts are required to be developed. Thus, if a project was to take  $x$  number of days, it could now take  $x-5$  days, which would result in a saving of 5 days in (manpower and other) costs, and provide further value addition to clients.

How are individuals assigned to specific roles? How are knowledge attributes assigned to individuals?

Individual roles are assigned by mapping the project requirements with the Competency and Learning Management System, which is an integral part of our (knowledge) portal. The system contains quantified competency levels, role suitability, role and individual profiles, re-evaluation reports, and the current and desirable competencies of the individuals; it sets individuals targets for personal development.

How has XYZ learned (more effectively) from its environment? Or, how is XYZ continuously learning? (Attempt to identify feedback).

XYZ learns from:

- industry best practices
- across the organisational (TATA) group; from other companies within the group but across domains; therefore has access to cross industry best practices
- cross industry best practices
- customers
- competitors
- visiting award winning organisations (in US)
- from suppliers/vendors
- institutes
- reports collated within the KMS (which depend upon the sensitivity if context)
- white papers published in the KMS

How does the portal help in identifying patterns for solutions to recurring problems?

By picking out best practices and lesson learnt. However, best practices have a positive connotation while lessons learnt are considered to have a negative connotation and are associated with post mortems, failed projects and project reviews.

Another aspect of identifying patterns is knowledge or artefact reuse.

Also, the customer complaints are logged in a systematic manner. After the specific problem is sorted out, the logs are analysed over a longer period for patterns. The effort is to identify the root causes of these problems and then proactively prevent recurrence. These patterns are shared both within the organisation and with the customer. In doing so, our organisation is able to advise the customer about potential problems and suggest appropriate solutions. This pattern seeking in customer logs also helps the organisation understand their customer's future requirements better, and translate this understanding to other customers with similar requirements.

---

17 Feb 2006

RG

XYZ is a CMM Level 5 organisation. Do you think that the processes in place for Level 5 accreditation provide sufficient support for XYZ's knowledge requirements, or are knowledge processes required beyond those present at Level 5? (Do they view knowledge (management) to be an integral part of CMM Level 5, or does it require another separate initiative)?

Level 5 accreditation at XYZ includes PCMM, where the people aspect is integrated into the processes and that includes the people knowledge requirements for the process. Learning programs for newly recruited employees address the PCMM Level 5 requirements. Training is planned for roles, and in certain training programs project leads and team members are merged. Training is mandatory and the emphasis during training is on processes as required during software development. Personal development programs for each individual are devised according to their skills and proficiency level to ensure a certain standard.

What is the methodology (Software Development Life Cycle) employed for the current projects being undertaken by you and your teams?

Scrum and Waterfall are the two methodologies being used by my teams.

Identify four important underlying constructs (that provide the foundation) of the software development process?

i) Very clear requirements. The client knows what they don't want but don't know what they want, which leads to rework. XYZ addresses this problem with a Traceability Matrix that maps and tracks the requirements and subsequent changes.

ii) Risks. These often manifest themselves when third parties (other than the Client and XYZ) are involved as suppliers, or outdated technology is used.

iii) Skill sets of the team members. This includes proficiency level, training and knowledge of individuals.

iv) Good testing to ensure quality.

What is the first thing for a new development project that you as a Project Manager do (or address)? (Are they knowledge related activities, for example, refer to the intranet portal; look to tap past project knowledge or experience; decide what tools to use for project implementation)?

The first thing is to look for people with the RIGHT experience for that particular type of project. This does not necessarily relate team members who will possibly implement the project but also seniors within the organisation who have worked on or have experience of similar projects in the past.

Thereafter, I would refer to ABC-portal with regard to best practices, metrics, project plan, time sheets for similar project profile codes.

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)? What tools are used to help decision-making, and what are the inputs and outputs of the decision-making process?

Decision-making plays an important role in:

- Estimation, which is a very grey process, and where past experience doesn't hold generally. New technology is required to be used, and people capability changes.

The SEPG and QAG (Quality Assurance Group) and ALL stakeholders help identify the project scope and problems. However, generally, it is the PM who estimates, while it is the team that develops the software. Therefore, the team should help decide during estimation.

- Risk Management: Decision-making helps in identify, prioritise and mitigate risks.
- Processes: during code-reviews within tight schedules, walk-throughs and Quality Assurance.
- Monitoring the project status and identifying what issues, if any, need to be addressed at the earliest and not too late in the development cycle.

What is the outcome of quality and testing?

Quality assurance is a part of the estimation process, as it takes up a substantial percentage of the project time, and needs to be included within the time schedule Gantt Charts. Quality assurance requires reviews and system walk-throughs to be conducted as part of the QA audits. Quality Control includes metrics and test plans to be included within the IPMS (Internal Project Management System). Defect prevention is the prime priority, and projects have Defect Prevention process owners. Projects conduct weekly meetings, and a DP report is provided to the Project Lead and DP process owner every fortnight, who further provide a month end report to the PM and Principal Consultants. The month-end report contains a consolidated causal analysis and a Pareto analysis. On the corporate level a report is submitted and reviewed every quarter, in which fresh issues/items regarding defect prevention are added to the overall DP checklist.

In your opinion, what plays a more important role in the development of software, Project Management or Software Engineering and why? (Which provides a bigger perspective of the developmental activities)?

In my opinion the two most important activities in software development are design and architecture. However, software development is a human activity and PM addresses the human aspect of the developmental process. I view the human aspect not as resources but as associates in the development process. PM helps me address the risk factors of losing personnel during critical times of the project schedule. For example, it helps me tackle a situation where a very important team member's wife is admitted to hospital just when the schedule is extremely tight and critical in terms of delivering a module. I would therefore rate PM as just above SE in terms of perspective.

How do your team members benefit from you and their knowledge and project experience?  
(How do your subsequent projects benefit from implementing previous/past projects)?

I view my role as a PM to be a role model and leader. I aim to benefit my team members through mentoring, as I feel that class room training does not help. Also, I delegate work with authority, and thus aim to make my team members more responsible and train them in decision-making.

I encourage my team members to observe while working with others as the most important learning is through observation. I observe deficiencies within my team members and aim to strengthen their weaker areas.

Subsequent projects benefit from experience gained in implementing past projects, best practices and lessons learned.

How does knowledge flow within your organisation, especially during the software development process?

Knowledge flows through the message corner, notice boards, email, ABC-portal and the XYZ monthly magazine.

-----  
JS

(PM)

2 Projects

36 + 2 Team members

Maintenance Projects for a US construction company's Group website; the project includes enhancements to the website.

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)? What tools are used to help decision-making, and what are the inputs and outputs of the decision-making process?

Decision-making occurs mostly in:

i) defect controlling

(Pareto analysis – find early)

Checklists to be followed.

Decision-making requires constant sharing of knowledge through mail, central repository, regular team meeting, social interaction. The tools used for DM include 6 Thinking Hats, Grid Analysis (metric based), Pareto Analysis, Fishbone, SWOT, Modelling, simulation, and refer to the Process Assets Library.

Central Reporting VSS – Visual source save version control cases.

*Basically I see my role as improving the process and making every person aware of their role in the process.*

Optimization to Level 5 of the CMM is not enough to cater to all the knowledge requirements. The 3 KPA's in Level 5 lay an emphasis on defining too much optimization, whereby *instead of maintaining the project, we get pre-occupied trying to maintain the processes.*

Do you think that the processes in place for Level 5 accreditation provide sufficient support for XYZ's knowledge requirements, or are knowledge processes required beyond those present at Level 5? (Do they view knowledge (management) to be an integral part of CMM Level 5, or does it require another separate initiative)?

Level 5 not enough

(3KPS'a)

Defining too much optimization (to many processes)

Instead of maintaining projects – they maintain processes.

PMR – Project monthly review Audit review meeting

Standardised tool based process – ultimately.

Client requirements and we (developers) need to merge.

Team Track – tool to manage teams and changes.

What is the first thing for a new development project that you as a Project Manager do (or address)? (Are they knowledge related activities, for example, refer to the intranet portal; look to tap past project knowledge or experience; decide what tools to use for project implementation)?

First thing I would do as Project Manager is

- i) try to ensure that I get the right people, right experience, right technology, for my project start-up by submitting my requirements in ABC-portal and MATC (Manpower Allocation Task Committee)
- ii) Perform a thorough analysis of the project and its requirements through face-to-face talks with clients onsite, preparing the SRS (Software Requirements Specification document), and reviewing it critically
- iii) Estimation through Function Point Analysis or any other new techniques, if available. Also, change requirements templates, if required.

- iv) Schedule preparation in which I listen to what the clients say and want, what is the time limit for the project, and what is my personal estimate about the time required.
- v) Team selection for implementation for which I discuss with technical experts, functional experts. In some cases, clients ask for team member resumes.

What is the outcome of quality and testing?

Quality Assurance (QA): quality is controlled by

- Defect prevention
- Requirements are reviewed by module expert
- Reviewing code and test cases
- Performing causal analysis on defect found and
- Defect Prevention meetings to review

Outcome :-

- Revision of DP checklist
- Coding standards
- Naming convention
- Delivery checklist

Handover – take over to be monitored by Group Lead.

How do your team members benefit from you and their knowledge and project experience?  
(How do your subsequent projects benefit from implementing previous/past projects)?

PM's knowledge and experience benefits team members through:

- Communication
- Interaction, both work and social
- Fortnightly meetings
- CCR -- Continuous capability improvement

In your opinion, what plays a more important role in the development of software, Project Management or Software Engineering and why? (Which provides a bigger perspective of the developmental activities)?

Software Engineering is needed for Project Management to succeed.

Identify four important underlying constructs (that provide the foundation) of the software development process?

Constructs

- (i) Well defined tool based processes that include defined roles and responsibilities
- (ii) Experienced team training programme and continuous learning

(Therefore, the interviewee identified people issues as important process constructs).

PM – IPMS (Integrated Project Management System) is a level 3 requirement.

---

PC

12-15 small projects

1 to 11 member per team

45 total

Methodology is Waterfall for development projects.

Identify four important underlying constructs (that provide the foundation) of the software development process?

Constructs

- i) Team composition
- ii) Good analysis & design
- iii) Estimation accuracy
- iv) Maintenance – good documentation and knowledge base;

Statistical Data

What is the first thing for a new development project that you as a Project Manager do (or address)? (Are they knowledge related activities, for example, refer to the intranet portal; look to tap past project knowledge or experience; decide what tools to use for project implementation)?

First Thing

- i) Estimation – FP (Function Point) method
- ii) Scheduling
- iii) Reserve pool for team selection and process
- iv) Identify the technology to be used



- v) Team composition – a mix of senior and junior members for analysis & design

Maintenance Tool – CR (Change Request) Previous experience

Projects are in Dotnet

I would also refer to Statistical data/Published guidelines for similar previous projects.

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)? What tools are used to help decision-making, and what are the inputs and outputs of the decision-making process?

Decision-Making usually occurs in:

- i) analysis and design; more in design – for example: How the ERD's design (Entity Relationship Diagram) translated into the database, and addresses future requirements and supports extensions. Tools – Grid method.

- ii) Testing phase:

- Break off for testing
- Unit testing
- Integration testing
- Demos

Tools -- Brainstorming

- iii) Change request -- whether to implement at a particular stage or not. Tools – Brainstorming, grid method, fishbone.

What is the outcome of quality and testing?

Outcome from testing

- Best learning
- Publish in portal (ABC-portal)
- Branch level library
- QAG (Quality Assurance Group) are owners (*pride and ownership*)

How do your team members benefit from you and their knowledge and project experience? (How do your subsequent projects benefit from implementing previous/past projects)?

Team members benefit from Project Managers knowledge and experience by:

- Grooming by hand holding (review & correct work) (sit on teleconference)

- Review comments
- Feed back

Do you think that the processes in place for Level 5 accreditation provide sufficient support for XYZ's knowledge requirements, or are knowledge processes required beyond those present at Level 5? (Do they view knowledge (management) to be an integral part of CMM Level 5, or does it require another separate initiative)?

I think whether it is CMM Level 5 or a separate knowledge process, we must lay an emphasis on:

- Quality processes
- Documentation
- Reviews
- Checkpoints
- Stringent processes

All of the above can be addressed by submitting Process Improvement Proposals (PIPs) in ABC-portal.

-----

AN

17/2

What is the first thing for a new development project that you as a Project Manager do (or address)? (Are they knowledge related activities, for example, refer to the intranet portal: look to tap past project knowledge or experience; decide what tools to use for project implementation)?

The first thing that I would do for starting a project is:

- Scoping – what are the boundaries, delivery, capability in terms of the project
- Estimation – tools etc.
- Budgeting
- Planning – Project Plan
- HR – team composition peer team at customer
- Stake-holders involvement plan: Who are the key stake holders
- HR for specific expertise: what training will be required

- Quality management
- Risk management

I would then have the above reviewed by the Quality Assurance Group and create IPMS model to execute the project. I would also refer to the Skill management system within our portal to identify local knowledge experts. (KMF)

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)? What tools are used to help decision-making, and what are the inputs and outputs of the decision-making process?

- i) Requirements stage – maximum DM or decision influencing (Use cases scenario mapping – tools)
- ii) Design – How to implement?  
What testing specifications?

Rational Rose – IBM

Master Craft – Internal to our organisation.

Inputs for DM

- Customer requirements – need to do and how they are to be accomplished.
- Availability requirements of the software system being developed.
- Performance requirements.
- Scalability requirements.
- Enhancements to the system being development.
- Previous experience.

Output of DM is the action taken iteratively.

Identify four important underlying constructs (that provide the foundation) of the software development process?

Constructs:

- i) How well the requirements have been captured
- ii) Quality management system: reviews
- iii) Team composition and individual skills
- iv) motivation

Do you think that the processes in place for Level 5 accreditation provide sufficient support for XYZ's knowledge requirements, or are knowledge processes required beyond those present at Level 5?

I am of the opinion that motivation is the key attribute that a software organisation requires past CMM Level 5 certification.

What is the outcome of quality and testing?

The outcome of quality and testing is increased reliability and performance of the software product. Also reflected in testing are:

- Training
- Estimation
- Architecture design
- Project Plan
- Project management deployed or implemented

For successful testing well defined test specifications are required that cover 100% of the product developed.

How do your team members benefit from you as a Project Manager and their knowledge and project experience?

Team Members benefit from Project Managers either formally during meetings, or informally during hands on sessions when we sit with them and explain issues. Team members also benefit within the learning development group when we conduct sessions that are published well in advance.

How do your subsequent projects benefit from implementing previous/past projects'?

An important factor to consider while referring to previous projects is if the projects met customer satisfaction after actual technical delivery.

Thereafter tools can be used elsewhere while artefacts can be reused in subsequent products. Also, we try to capture experience through Best Practices and learning's in the EKMS. Knowledge and learning in projects are further captured by recording and documenting problem solving within PAL (Process Asset Library).

How does knowledge flow within your organisation, especially during the software development process?

Knowledge flows:

- Via customer
- Delivery
- Standard bodies
- Conferences – white paper
- Top down through subject matter experts

In your opinion, what plays a more important role in the development of software, Project Management or Software Engineering and why? (Which provides a bigger perspective of the developmental activities)?

Definitely Project Management because it provides a bigger perspective of the development process.

-----  
PB

What is the knowledge management initiative at XYZ and what is the vision behind it?

ABC-portal -- Automation Framework – has been in full use for the past 4 years but subsystems existed earlier. The framework evolved over time rather than come up as a one-time exercise/initiative, and is basically an integration of developmental, project management knowledge activities. Knowledge Management existed before (historically) also within the organisation but was previously viewed differently, for example a Process Assets Library (PAL), or Case-Based Tools (CBT), available in the physical library.

How does knowledge flow within your organisation?

Knowledge within our organisation, flows bottom-up with regard to technology, top-down with regard to policies and strategies, and also laterally in a chaotic, flexible manner without constraints.

The medium for knowledge to flow is within our tools, email, notice boards, magazines and mail corner.

Also, our organisation has interaction at all levels. This interaction could be either formal or informal, and in a structured manner. Unstructured interaction does exist, but cannot be accounted for (taken into account).

What is the first thing for a new development project that you as a Project Manager do (or address)? (Are they knowledge related activities, for example, refer to the intranet portal; look to tap past project knowledge or experience; decide what tools to use for project implementation)?

The first thing that I address is from the project management perspective and that is to study the contract with the client. Next I decide upon my project team and conduct a start-up meeting where we define the ground rules for the project. In this meeting we also decide the technologies to be used to implement the project.

Identify four important underlying constructs (that provide the foundation) of the software development process?

The important constructs are people, tools and methodology.

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)? What tools are used to help decision-making, and what are the inputs and outputs of the decision-making process?

- Planning,
  - estimations,
  - reviews,
  - testing (*very important*), and
  - customer interaction (*important*).
- 

BC

PM

2 projects

20 team members

Scrum which is release based deliveries, and Maintenance projects.

What is the first thing for a new development project that you as a Project Manager do (or address)? (Are they knowledge related activities, for example, refer to the intranet portal; look to tap past project knowledge or experience; decide what tools to use for project implementation)?

The First Thing I do when I am awarded a project to execute is study and prepare the:

- Contract
- Proposal
- Requirements
- Estimation
- Implementation

I then conduct Project Start up meetings to form support groups, and get team commitments.

Thereafter another important activity is planning resources, how to execute the project, and a unified project plan.

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)? What tools are used to help decision-making, and what are the inputs and outputs of the decision-making process?

The decisions made during the requirements and design stages are extremely important, while other important decisions are also made during implementation.

Identify four important underlying constructs (that provide the foundation) of the software development process?

An important construct is Project Management.

In your opinion, what plays a more important role in the development of software, Project Management or Software Engineering and why? (Which provides a bigger perspective of the developmental activities)?

Initially I think that software engineering provides a more important perspective, but as you begin to take on more responsibilities, then Project Management assumes more importance in the latter stages.

What role does Knowledge Management play in implementing your software projects?

Knowledge management plays a supporting role while developing software projects. This support is provided by the Assets library and Quality Management Systems (QMS) process handbooks, guidelines and checklists. All our activities use the intranet portal *ABC-portal which is the life-line of our organisation*. *Do not think beyond the portal --- assume that it is the complete answer to all their knowledge requirements, and that no other knowledge support is required.*

Do you think that the processes in place for Level 5 accreditation provide sufficient support for XYZ's knowledge requirements, or are knowledge processes required beyond those present at Level 5?

Well defined processes are a part of Level 5 and are sufficient.

What is the outcome of quality and testing?

The outcome of quality and testing is to:

- Log defects into QMS
- Perform causal analysis and enter the findings into the IPMS
- Establish Best practices and lessons learnt, and publish them in the Asset library

How do your team members benefit from you and their knowledge and project experience?

Team members benefit through

- Mentoring
- Communication
- Good practices

How does knowledge flow within your organisation, especially during the software development process?

Knowledge flows within our organisation through:

- the Asset Library
- knowledge sharing sessions -- Expertise sharing
- Training plan for each month (Classroom and for CBT)
- Mandatory training – IKMS, defect prevention, Causal analysis



AK

PM

One Project

12TM

The project is a Testing project for a KM company. The project remit is to test and analyse an intelligent search tool. Therefore the project is context based and after analysing the functionality of the product, we validate its basic intentions and give suggestions regarding the product.

What is the first thing for a new development project that you as a Project Manager do (or address)?

The first things I address regarding a project are:

- Requirements
- Background of Client
- Schedule

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)?

Contract negotiations and initial planning.

Identify an important underlying construct (that provides the foundation) of the software development process?

An important construct for software development would be the core team.

Do you think that the processes in place for Level 5 accreditation provide sufficient support for XYZ's knowledge requirements, or are knowledge processes required beyond those present at Level 5?

Level 5 provides us with well built and integrated processes, with knowledge management aspect already addressed.

How do your team members benefit from you and their knowledge and project experience?  
The team members benefit through communication and involvement that is provided via email, teleconferencing.

How does knowledge flow within your organisation, especially during the software development process?

Emails, formal, sessions, informal interaction, internet, intranet portal – ABC-portal.

How do your subsequent projects benefit from implementing previous/past projects?

Subsequent projects benefit through the inherent experience we gain while implementing projects and capture in our documentation.

Team meetings, minutes of meetings with clients in true spirit capture thoughts and issues which help in subsequent projects.

I also think that blogs on our website help us in implementing projects. *Important.*

-----

PJ

One project in addition to internal development.

7+4 = 11 team members

Scrum -- Iterative release of versions

What is the first thing for a new development project that you as a Project Manager do (or address)? (Are they knowledge related activities, for example, refer to the intranet portal; look to tap past project knowledge or experience; decide what tools to use for project implementation)?

- Requirements document, looking at the first iteration of scrum
- Training of tools and technologies
- Training of Components
- Revisit Requirements for second iteration

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)? What tools are used to help decision-making, and what are the inputs and outputs of the decision-making process?

- Planning, budgeting
- Resources

- Important decisions are also required to be when team members resign or leave the project, or when the product is rejected

Identify four important underlying constructs (that provide the foundation) of the software development process?

- Skills
- Clear requirements
- Right tools
- Constant Stakeholder/customer interaction

Do you think that the processes in place for Level 5 accreditation provide sufficient support for XYZ's knowledge requirements, or are knowledge processes required beyond those present at Level 5?

CMM Level 5 certification is not enough to cater to the knowledge requirements of a software project, as Level 5 only provides guidelines for implementing the project.

Benefits of past experience of implementing projects can help subsequent projects by:

- having the right tools to record experience
- Summary reports or white paper to capture experience
- Training sessions based on experience

What is the outcome of quality and testing?

- Causal analysis
- Logging the findings of the causal analysis into the system
- Future employees need to study the causal analysis findings in the system during their training

How do your team members benefit from you and their knowledge and project experience?  
(How do your subsequent projects benefit from implementing previous/past projects)?

Team members benefit by:

- Empowering them to perform their tasks
- taking risks
- facing problems and solving them
- training
- feedback

- mentoring

Subsequent projects benefit through:

- Experience
- Improving Processes, planning, thought processes

If a Project Manager leaves the organisation it is imperative to have team members interact with him/her to ensure that all important issues are taken into account. Further contact information of the person leaving must be available for any clarification required.

Such information and experience must be logged back into the system.

How does knowledge flow within your organisation, especially during the software development process?

- Customer interaction
- Reviews (including peer reviews)
- Email
- Meetings
- mentoring
- ABC-portal, upon which we rely for administration and resource management, and logistics
- IPMS for PM into which improvements are updates are made

Also, I would like to mention that constant interaction and reviews help in managing the knowledge within the organisation.

In your opinion, what plays a more important role in the development of software, Project Management or Software Engineering and why?

PM and SE are both equally important in perspective.

-----  
RA

Group Lead (Senior)

3 projects

11+2+7 team members

Standard process – IPMS

Methodology – Tasks, Time Sheets, Track reviews

Iterations - 1 month

Testing project

Which stage of the software development process do the important decisions occur (where 20% of the decisions affect 80% of the project outcome)? What tools are used to help decision-making, and what are the inputs and outputs of the decision-making process?

The important decisions in a software project are made during the:

- Design stage regarding the choice of methodology and
- Requirements stage

Identify four important underlying constructs (that provide the foundation) of the software development process?

- Processes in place (process framework)
- Best practices

Do you think that the processes in place for Level 5 accreditation provide sufficient support for XYZ's knowledge requirements, or are knowledge processes required beyond those present at Level 5?

CMM Level 5 has built in statistics and guidelines to help implement a project, for example, timelines which help prepare project plans.

What is the outcome of quality and testing?

- Check lists are updated, and
- Learn from captured bugs and mismatches identified

How do your team members benefit from you and their knowledge and project experience?  
(How do your subsequent projects benefit from implementing previous/past projects)?

Team Members benefit by

- Managing / tracking their activities
- Gaining Technical experience

Subsequent projects benefit with increased awareness of possible problems and outcomes especially on similar projects.

How does knowledge flow within your organisation, especially during the software development process?

- communication
- personal interaction
- e-mail
- telephone
- follow up

A suggestion I would like to make is that asset libraries need to capture experience of planning, estimation and client feedback

In your opinion, what plays a more important role in the development of software, Project Management or Software Engineering and why?

PM and SE are both equally important in perspective.

-----

## Appendix 5 Sample Questionnaire

### Knowledge Management Support for Software Projects

The questionnaire provided below is part of a research project being conducted at Middlesex University, UK that addresses the need to provide knowledge management support to software projects. Please provide short open-ended responses, and the two page questionnaire would require 20 minutes to complete. The name of the organisation and all responses will be maintained confidential and anonymous.

1) Name:

2) Designation:

3) Methodology of Project:

4) List five important underlying constructs of the software development process (Example: team composition, requirements, quality, project management etc).

i)

ii)

iii)

iv)

v)

4 a) What tools, if any, used to aid the above mentioned constructs.

i)

ii)

iii)

iv)

v)

5) Where within the software processes are important decisions required which affect the outcome of the project?

5 a) What are the inputs for these decision?

b) What tools are used to aid this decision making?

- 6) What is the outcome of the defects detected and corrected during testing?
  
- 7) How do team members benefit from each other's knowledge and experience?



1) Name: G S

2) Designation: Project Manager

3) Methodology of Project: Big Bang

4) List five important underlying constructs of the software development process (Example: team composition, requirements, quality, project management etc).

i) Cost

ii) Planning

iii) Synergising team goals into final goal

iv) Quality control

v) Managing time scale

4 a) What tools, if any, used to aid the above mentioned constructs.

i) Budgeting

ii) Thorough understanding of objective

iii) Good flow of information through individuals/teams

iv) Keeping sight of the objective

v) Ensuring 'buffers' within cost/time/teams

5) Where within the software processes are important decisions required which affect the outcome of the project?

At the planning stage

5 a) What are the inputs for these decision?

A thorough understanding of the final outcome and processes involved

b) What tools are used to aid this decision making?

Feedback from end users/managers of teams that are/may be involved in development

6) What is the outcome of the defects detected and corrected during testing?

Some defects are inherent shortcomings in software and have to be worked around rather than worked at. Some defects were detected as a results of end user testing and rectified by

developers. Few 'defects' cropped up as a result of improper/incorrect information received during planning.

7) How do team members benefit from each other's knowledge and experience?

Since different skills and qualities are required to be brought together into making a comprehensive piece of software, it was found that 'bouncing of ideas' between teams worked best. Working in isolation (individually or in teams) was discouraged and teams were made aware of each others limitations, skills and procedures.

8) How do subsequent projects benefit from the experience gained while implementing past projects?

The knowledge, experience and cohesion of the teams during the process is extremely valuable in future projects. While individual skills are important and encouraged, what is important is that the entire team achieves the set objectives. After a few projects have been implemented, the power of 'repetitiveness' sets in and is best case scenarios, the different elements come together as a well-oiled piece of machinery.

---

1) Name: D S

2) Designation: Quality assurance manager

3) Methodology of Project: Bottom up

4) List five important underlying constructs of the software development process (Example: team composition, requirements, quality, project management etc).

i) Quality

ii) risk management

iii) well defined process

iv) defect prevention

v) defect fall back positions

4 a) What tools, if any, used to aid the above mentioned constructs.

i) lotus notes

ii) master craft

5) Where within the software processes are important decisions required which affect the outcome of the project?

testing

5 a) What are the inputs for these decision?

Use case specifications

b) What tools are used to aid this decision making?

6. What is the outcome of the defects detected and corrected during testing?  
Usability and functional defects

6. How do team members benefit from each other's knowledge and experience?  
Problem isolation

6. How do subsequent projects benefit from the experience gained while implementing past projects?  
knowledge acquired

9) How do subsequent projects benefit from the experience gained while implementing past projects?

---

1) Name: M L

2) Designation: Project integration manager

3) Methodology of Project: Spiral

4) List five important underlying constructs of the software development process (Example: team composition, requirements, quality, project management etc).

i) Project design

ii) Planning

iii) Quality process

iv) requirements

v) module breakdown and integration plan

4 a) What tools, if any, used to aid the above mentioned constructs.

i) Rational rose

ii) lotus notes

5) Where within the software processes are important decisions required which affect the outcome of the project?

testing

5 a) What are the inputs for these decision?

customer requirement

b) What tools are used to aid this decision making?

email

7. What is the outcome of the defects detected and corrected during testing?  
project plan

7. How do team members benefit from each other's knowledge and experience?  
better planning,

7. How do subsequent projects benefit from the experience gained while  
implementing past projects?  
Better software designs, bug fixing

---

1) Name: S B

2) Designation: Project Manager - Customised Billing Solutions

3) Methodology of Project: Spiral

4) List five important underlying constructs of the software development process  
(Example: team composition, requirements, quality, project management etc).

i) Team composition

ii) Project Management

iii) Structured milestone plan with clearly defined milestone deliverables

iv) HCI user adaptability

v) Requirements

4 a) What tools, if any, used to aid the above mentioned constructs.

i) Rational Rose

ii) Microsoft Project

5) Where within the software processes are important decisions required which affect the  
outcome of the project?

Requirement analysis, Design and milestone accomplishment

5 a) What are the inputs for these decision?

Customers requirements, user's usability expectations, design and milestone deliverables

b) What tools are used to aid this decision making?

Rational rose and project

8. What is the outcome of the defects detected and corrected during testing?

Design modifications, although we try to keep the design changes right down. Impact of the modification is accessed on the whole system and the changes are made throughout the system to minimise mismatch issues on implementation.

8. How do team members benefit from each other's knowledge and experience?

Experience does help in understanding problems and creating effective solutions.

8. How do subsequent projects benefit from the experience gained while implementing past projects?

Bug fixing and streamlined designs