Requirement Driven Knowledge Management System Design to Support Automotive Product Development

Pengcheng Zhang

A thesis submitted in partial fulfilment of the requirements of the University of Greenwich for the Degree of Doctor of Philosophy

October 2011

DECLARATION

"I certify that this work has not been accepted in substance for any degree, and is not concurrently being submitted for any degree other than that of Degree of Doctor of Philosophy *being studied at the University of Greenwich. I also declare that this work is the result of my own investigations except where otherwise identified by references and that I have not plagiarised the work of others".*

Signed by *Pengcheng Zhang*

____

Prof. James Gao

____

Acknowledge

Firstly I would like to thank my first supervisor Professor James Gao for his help, guidance, encouragement and support throughout the research project. I would also like to thank my second supervisor Professor Predrag Rapajic for his concern and help.

Secondly, I would like to thank the vice-president and other members of staff in the collaborative company for their support and advice on data collection, requirements capturing, business and knowledge capturing, verification, and evaluations.

Thirdly, I would like to thank all colleagues from the "Centre for Innovation Product Development" of Engineering School at University of Greenwich for their help, support and suggestion.

I would also like thank Dr Xinmin Jin for his generous help and advice.

I would like to thank all staff at School of Engineering at University of Greenwich for their support, especially Dr Steve Woodhead, Prof. Ndy Ekere, Mr Ian Cakebread and Dr Dele Owodunni.

Finally, I would like to thank fully support from my wife Su Liu and my parents. I can not finish my PhD without their support.

Abstract

Nowadays, New Product Development (NPD) has become a business priority in manufacturing companies due to international competition in terms of meeting higher and changing customer requirements, generating high profit at low cost, and maintaining sustainable development and growth. Through literature review and industrial investigations, it has been recognised that NPD is an information and knowledge intensive process. However, in current practice, enterprise knowledge is not properly managed or easily accessible. Many service providers have not followed the good practice of considering business objectives and end users' requirements as main drivers of knowledge management system development and implementation.

This doctoral thesis presents a methodology for the design and development of Knowledge Management (KM) systems to support NPD based on Enterprise Architecture Frameworks (EAFs). The project focuses on IT system specifications generation driven by business and knowledge users' requirements in the automotive industry. Current EAFs have been developed by researchers and practitioners to help enterprises to design their information systems based on business objectives and user requirements. However, these frameworks are mainly proposed to manage information and data such as finances, resources, management and engineering documents, not for the increasingly important enterprise knowledge, especially tacit and unstructured knowledge.

This project aims to extend the capabilities of the latest enterprise architecture frameworks so that not only data and information, but also enterprise knowledge can be managed. A guideline in the form of a flowchart has been developed, which provides a process that can be followed and used by system developers and implementation. The extended EAF has been implemented as easy-to-use folders for the development of a structured knowledge base. A case study in an automotive company proved that the methodology can be used to produce the functional specifications of their IT systems to include knowledge management capability. The system specification can then be used, either to assess a company's existing information systems and direct its future system development and implementation; or to develop/implement a complete new information system from scratch.

Publications

Journal Papers

1. Zhang, P., Gao, J., and Wang, Y., “Requirement Driven Knowledge Management System Design to Support Automotive Product Development”, *International Journal of Product Lifecycle Management, accepted.*

2. Zhang, P., Gao, J., Jin Xinming and Yinglin Wang, “A Requirement Driven Framework for the Design of Information/Knowledge Systems to Support Product Development”, *International Journal of Manufacturing Systems, accepted.*

Refereed Conference Papers

1. Zhang, P., Baxter, D.I., Gao, J., and Kalta, M., “Developing a design process model for the reuse of knowledge*”, Proceedings of DET2007 4th International Conference on Digital Enterprise Technology,* 19-21 September 2007*,* Bath, United Kingdom

2. Zhang, P., Gao, J., and Jin, X., “Requirements Driven Knowledge System Design for New Product Development Using Enterprise Architecture Methodologies”, *The 7th International Conference on Manufacturing Research (ICMR09),* September 8-10, 2009, University of Warwick, UK,

3. Zhang, P., and Gao, J., “Requirements Driven Knowledge System Design for Product Development”, *Proceedings of DET2009 6th International Conference on Digital Enterprise Technology*, 14-16 December 2009, Hong Kong

Glossary

ADM – Architecture Development Method

API – Application Programming Interface

CAD/CAM – Computer-Aided Design and Manufacturing

CAPP – Computer Aided Process Planning

CMS – The Center for Medicare and Medicaid Services (CMS) of U.S. Department of Health and Human Services

DoDAF – Department of Defense Architecture Framework

CPM – Critical Path Method

DR – Design Roadmap

DSM – Design Structure Matrix

EAF – Enterprise Architecture Framework

EBA – Enterprise Business Architecture/

EIA – Enterprise Information Architecture

ETA– Enterprise Technology Architecture

ER – Entity Relationship

ERP – Enterprise Resource Planning

FEAF – Federal Enterprise Architecture Framework

GEAF – Gartner Enterprise Architecture Framework

IDEF –Integrated Computer Aided Manufacturing (ICAM) Definition

III Reference Model – Integrated Information Infrastructure Reference Model

IT – Information Technology

KBE – Knowledge Base Engineering

KM – Knowledge Management

MOKA – Methodology for Knowledge Based Engineering Applications

NIST – National Institute of Standards and Technology

NPD – New Product Development

NVH – Noise, Vibration, and Harshness

OA – Office Automation

PERT – Project Evaluation and Review Technique

PLM – Product Lifecycle Management

RAD – Rapid Application Development

RM – Requirement Management

SADT – Structure Analysis and Design Technique

SDLC – System Development Life Cycle

SME – Small and Medium Enterprise

SOA – Service-Oriented Architecture

SUV – Sport Utility Vehicle

TOGAF – The Open Group Architecture Framework

UML – Unified Modelling Language

Table of Contents

List of Figures

Chapter 1: Introduction

This chapter aims to give a general overview of this research project. Firstly, industrial background of this project is described. The problems facing current industry practice are identified and discussed. According to the identified problems, the research area and focus are decided. The aim and objectives of the research area are then defined. Then currently available technologies in the research area are introduced. The research scope and research methodology are also described. The potential benefits of this research are briefly introduced. Finally, the background of the collaborating company is introduced.

1.1. Industrial Background

Since the Industrial Revolution in the middle of 19th century, manufacturing industry becomes the most significant element in global economy (Mantoux, 1961). During the past 150 years, industry experienced three main innovations, i.e., the first industrial revolution was in 18th century with mechanisation in Britain (Brown, 2003), the second industrial revolution was from 1865 to 1900, to raise industrial power of manufacturing in the United Kingdom (Bernal, 1970), and the third industrial revolution began in the middle of 20th century within atomic energy, computer and space technology in the United States (Hughes, 1989). The common aim of these three revolutions is seeking the maximum productivity with breakthrough technologies, in order to achieve maximum profit.

After the third industrial revolution, the main aim of industry is gradually transferred to reducing cost and increasing efficiency rather than seeking pure technology development. It has been recognised that controlling cost is another way to achieve the maximum profit. For example, Henry Ford innovated the assemble line for the automotive industry to help save cost and improve efficiency in the factory. This innovation has already become the main adoption of most industries (Norcliffe, 1997). Usually, many factors determine cost directly, such as material, labour, risks or mistakes in the production process and some integrant factors like plant and transportation cost. Just controlling cost is not enough to survive in the global market. Product quality is another important issue. Better product quality with low costs is the goal for every manufacturing company. Many technologies are developed to support cost reduction and quality assurance, such as logistic management and Information Technology (IT).

Nowadays, new product development has become very important part in manufacturing companies due to international competition in terms of cost, quality and delivery, and meeting customer requirements. Therefore, it is the focus of the international research community to develop innovative methodologies to support the new product development process. Product development process not only decides whether the new product is successful, but also determines the future directions of the enterprise and its competitive advantages. Actually, the ultimate goal of each business is to achieve the maximum profit for the company. In manufacturing companies, product innovation or new product development (NPD) usually leads to the largest part of the cost (Dallas, 2006). Therefore, the high cost of product development is a serious problem for manufacturing companies. There are unnecessary expenditures and unsuccessful product developments existing in the NPD process (Crawford, 1979).

Recently, enterprise information, especially knowledge is recognised as the most important asset which influences NPD directly. However, knowledge is the most uncertain and difficult part in the NPD process. Unmanageable knowledge increases the unsuccessful rate of the NPD, and it leads to unnecessary costs and unpredicted situations in NPD process (Marbán et al, 2008). Especially, product knowledge is very difficult to clarify. Polanyi (1966) stated that “we know more than we can tell”. This statement represents the obvious characteristics of knowledge. Advanced knowledge management (KM) technologies are needed to manage product knowledge in order to control the whole new product development (Liebowitz, 1999; Riege, 2005).

Before research and development effort was devoted to knowledge management, significant resources have been focused on information systems development to support new product development. It has been recognised that IT can help organisations better manage human resources, business resources and technology resources (Powell and Dent-Micallef, 1997). It has already been implemented broadly in industry. However, in the current global innovation environment, data and information can not satisfy industrial requirements in current global competition. Therefore, knowledge becomes more and more important in global manufacturing business.

Enterprise knowledge has been recognised as one of the most important competitive factors of business success (Liao, 2003). Most, if not all, enterprises have already implemented information and communication systems to help manage enterprise information, and a growing number of enterprises are investing in enterprise knowledge management (Lee and Choi, 2003). In manufacturing, product related knowledge is regarded as a necessary connection between different product generations and similar products (Sharma and Gao, 2006). Effective capture and use of knowledge is vital to successful product development which determines the future direction of an enterprise and its competitive advantages (Ichijo and Kohlbacher, 2008). However, in current practice, knowledge is still not properly captured, maintained or easily accessible. There is a strong industrial need for better managing enterprise knowledge, and the design of a knowledge management system should be driven by corporate objectives and knowledge users' requirements (Richardson et al, 2006). An effective and efficient KM system helps manufacturing companies to gain maximum profits in the global marketplace (Stokes, 2001).

Current information systems can not satisfy industrial requirements for managing enterprise knowledge. There are no formal methodologies to follow when enterprises are planning, designing and implementing KM systems. Attempts to finding a satisfying technology have been made by researchers and practitioners. A formal framework which can be used to guide the design and implementation of knowledge management systems may be proposed on the basis of the widely used Enterprise Architecture Frameworks (EAFs).

Most information systems in use are designed or implemented without systematically analysing corporate objectives and business requirements, and without following good formal methodologies (Kuang and Gao, 2006). It is common that various information systems used in the same company and in the same product development team do not communicate well each other. Therefore, Enterprise Architecture Frameworks (EAFs) have been proposed to help organisations address the complex system and business alignment problems in information system design and implementation (Sessions, 2007). As knowledge, which is often less structured and difficult to define, becomes more important to the globalised manufacturing business, current information systems should be extended to manage enterprise knowledge, and hence the authors explored the potential of using EAFs to design information systems with knowledge management capabilities. The corporate objectives and the requirements of knowledge users in the product development process, i.e., all members of the product development team and related personnel, are used as the main driver of system design.

1.2. Proposed Research Area

According to the problems described above, this research project proposed to develop a methodology for the design and development of a knowledge management system for manufacturing enterprises based on some methodologies with real industrial case studies. It has been identified that there is a strong need for better managing knowledge in the new product development process. Many information/knowledge management systems are designed without systematically analysing corporate objectives and knowledge users' requirements, and without following formal requirement driven enterprise information system design methodologies.

Enterprise Architecture Frameworks have been proposed and developed to help enterprises design their information systems based on corporate objectives and user requirements. However, EAF have not given sufficient consideration of how to manage enterprise knowledge. This project aims to extend the newly developed enterprise architecture frameworks so that not only information, but also enterprise knowledge can be managed. This project focuses on the new product development process, and thus the enterprise in this context is defined as the product development department or organisation. The potential benefit of this research is that manufacturing companies may use the developed methodology of this research to produce the system specifications based on the corporate objectives and knowledge users' requirements. The system specification can then be used to assess a company's existing information systems, and direct the company's future system development and implementation.

1.3. Available Technologies

Currently, many manufacturing companies use knowledge management for specific problems solving (Vestal, 2005). Knowledge based engineering (KBE) is a technique to help organisations to transfer and maintain the useful knowledge in the product development and manufacturing processes. A example methodology of KBE was developed by MOKA Consortium which is Methodology for Knowledge Based Engineering Applications (Stokes, 2001). MOKA provided a methodology consisting of the meaning of KBE, a process to achieve the models and software to support it. Chua (2004) suggested that a comprehensive KM system should consider the fundamental KM processes such as knowledge acquisition, transferring, storing and sharing. He suggested that this representation is suitable for social, economic and technological factors of an enterprise. Jung et al (2006) reported an integrated architecture for knowledge management systems and business process management systems based on process-oriented knowledge management. In the initial stage, the business process management lifecycle and the knowledge management lifecycle are integrated, as the two lifecycles have similar structures and steps. This integration can help enterprises develop their knowledge structure based on in-house business processes.

Information Technology (IT) systems have been more widely and successfully implemented in business, comparing with the less mature KM systems (Burrows et al, 2005). Chen et al (2003) suggested that traditional IT systems are not designed to support organisation learning. They are normally passive and straightforward. Information is only provided when users ask for it. They do not provide proactive communication in an enterprise. They normally provide a single port of accessing databases, and this does not satisfy the real working environments of most enterprises. In the current industrial circumstances, a desired system is like a workshop which can manage communication, working relationship, people behaviours, knowledge idea, culture and context of enterprise (Ruggles, 1998, Bush and Tiwana, 2005). Enterprise Architecture Frameworks (EAFs) are developed to help address the complex systems and business alignment problems in information system design and implementation (Sessions, 2007).

Product lifecycle management (PLM) is another technology to manage NPD in a systematic way. From product development point of view, PLM is a process of managing the whole lifecycle of product development consisting of its conception, design, manufacture, service and maintenance (CIMdata, 2009). From the PLM system point of view, PLM systems integrate people, data, processes and business systems and provide a product information backbone for companies and their extended enterprises (PLM Technology Guide, 2009). Broadly speaking, PLM systems manage the creation, mortification and exchange of product information

(Srinivasan, 2008). However, all these technologies have different kinds of weaknesses and limitations. For example, PLM and EAF do not have sufficient capability to manage knowledge. Therefore, it is a strong requirement for better managing knowledge in NPD considering both knowledge user's requirement and business objectives.

Therefore in this project, requirement management is vital as a technology tool. When the main business of a product development company is product development, the business requirement will become to product focused. It needs an appropriate way to model product development process. Business requirement lifecycle also needs to consider how to manage its generated knowledge and then how to reuse the knowledge. Product requirements come from the existing customers, new markets, business requirements and relevant policies. Therefore, a comprehensive product development process driven by product requirement is necessary in order to better manage product knowledge during the product development process.

Knowledge user's requirement is the main driver of KM system design in this project. Therefore, using an appropriate method to capture knowledge user's requirement is vital. The knowledge user's requirement is mainly used to develop system specifications. As a large system development project, not all users' requirements will be directly incorporated into system specifications. Some users' requirements are rarely or never used in real-life applications. Therefore, requirement prioritisation will be used to determine what kinds of knowledge users' requirements will be transferred to system specifications.

In this project, not only knowledge user's requirements for IT/KM system are captured, but also the actual relevant knowledge needed is captured. Therefore, the whole process of capturing and developing knowledge user's requirement is very complex. There is a requirement for using a formal process which can be followed by system developer who firstly uses the proposed methodology. This process can be defined as system development life cycle (SDLC) which is the core of any system development. It is a process for creating system which system developers can use it to develop their systems (Blackchard and Fabrycky, 1990). System development life cycle can help organisations to deal with their complex system development problems. However, there is not a system development life cycle for developing KM systems.

Most current system development life cycles do not cover the whole KM system development. Therefore, the proposed methodology needs to identify and provide a system development life cycle process for users.

The proposed system development methodology also needs to define its structure of methodology. There are 5 types of system development methodology: Waterfall, Prototyping, Incremental, Spiral and Rapid Application Development (RAD). All these 5 structures of system development methodologies have their advantages and weaknesses. After comparing them, the structure of the Prototype methodology is suitable for this project. It provides an iterative structure for requirements definition, system design, coding and testing, in order to identify a comprehensive user's requirements and system specifications through the iterative investigations based on the very complex KM system development. It should be noted that this project does not follow the complete process of the Prototype methodology as coding and implementation is outside the scope of this project. However, users of the methodology developed in this project are recommended to follow the Prototype methodology for system development. More information about the Prototype methodology will be provided in chapter 2.

There are many processes involved in this project, such as business process in the collaborating company. Therefore, an appropriate process modelling technique can help this project to manage all processes, particularly for product development process. Design Roadmap is adopted as the modelling techniques. It contains both activities (tasks) and objects (features). This characteristic makes it easily represent inputs, outputs and activities. It makes the process very easy to understand. It can deal with both simple processes and complex processes. Therefore, it is suitable for this project. More information will be provided in the literature review chapter.

1.4. Research Aim and Objectives

The aim of this research is to develop a formal methodology for knowledge system design to support new product development based on user requirements and business objectives. The main research objectives are:

- To investigate knowledge involved in new product development, business process and organisational structure of automotive manufacturing enterprises,
- To identify and classify requirements of different stakeholders and knowledge user's requirements in the new product development process,
- To propose a methodology for the design and the development of knowledge management systems based on formal enterprise architecture frameworks,
- To develop system specifications for a knowledge management system design using the proposed methodology and
- To verify and evaluate the knowledge management system design methodology using industrial examples

This project proposes requirement driven knowledge management system design methodology to support automotive product development. The methodology consists of a guideline (in the form of a flowchart), an improved Enterprise Architecture Framework (EAF) based on TOGAF, and a "Folder-based EAF Implementation". The improved EAF provides the principles of how this proposed methodology works. The guideline provides a process which can be followed and used by system developers. The "Folder-based EAF Implementation" provides a structured knowledge base to support the guideline.

The methodology should be appropriate for both information system development and knowledge management system development. The methodology is based on both business objectives and knowledge users' requirements, and provides a formal process for enterprise knowledge management system design. An Open Group Architecture Framework has been enhanced, extended and used as the basis of the proposed methodology. The main advantage of the proposed methodology is that it emphasises knowledge management, whilst existing enterprise architecture frameworks are mainly for information management. This methodology is being verified and evaluated with real industrial case study.

Manufacturing companies may use the methodology to produce the system specifications of their IT systems to include knowledge management capability. The system specification which is the final output can then be used, either to assess a company's existing information systems and direct its future system development and implementation; or to develop/implement a complete new information system from scratch.

1.5. Research Scope

The scope of the investigation is in the manufacturing sector. The research project focuses on complex products, such as automotive cars. The main area of the focused is the new product development process. Although this project relates to the production process, it focuses on the early stages of the NPD process. The final output of the research is the system specification and its functional requirements of the system with verified examples. The actual development of the software system of the KM system is out of the scope of this project. The reason is that the resultant system specification can be used for assessing a company's existing information systems, as well as for new system installation and complete new system development. The methodology and tool are considered to apply in the whole NPD process in the industrial environment. An automotive manufacturing company has been the collaborator of this project and provided examples for case studies to test the proposed methodology.

1.6. Research Methodology Adopted in this Project

Each research project has to follow an appropriate research methodology in order to ensure that the research project can be finished successfully. Before defining the terminology "Research Methodology", the terminology "research" needs to be defined. "Research" is defined as a procedure to find one attempt systematically with the supported facts, in order to answer the questions or to solve problems (Leedy, 1989). The terminology "research methodology" can be explained simply as the method of how researchers conduct their research. A comprehensive definition of research methodology is defined by Pattron (2000) as “a highly intellectual human activity used in the investigation of nature and matter and deals specifically with the manner in which data is collected, analysed and interpreted. From this definition, research methodology can be viewed as a method or a technique which can be followed in research projects, in order to solve problems through a set of human activities such as observations and experiments.

The first step of defining research methodology is to identify the research types. Kumar (2011) stated that there are three perspectives to classify research. The first perspective is the application of research findings. Research can be divided into two types: pure research and applied research. Pure research is that the theories and hypotheses are developed by researchers intellectually. It may or may not have a practical application currently or in the future. Applied research is the research applied to the collections of information which can support and verify from several aspects. In other words, the applied research is established based on real problems and finding a solution to solve these real problems. Kumar (2001) stated that the most of the current social sciences and engineering research projects are applied research.

The type of this research project is applied research. According to previous discussions, the problem facing current industry is that there is not a formal methodology for design and development of knowledge management system which considers both user’s requirements and business objectives. This research seeks to develop a formal methodology to solve this problem.

The second perspective is the objectives of the study. Research can be divided into four types based on the research objectives: descriptive, exploratory, explanatory and correlational study. Descriptive research is to describe a situation, a problem, or a phenomenon in a situation. For example, it may attempt to describe the current situation of the product development in an automotive company. Correlational study research is to discover the relationships or links between two or more aspects. For example, it may attempt to find the relationship between product knowledge and cost of product development. Explanatory research is to find the reasons and methods between two aspects. For example, it attempts to explain why and how product knowledge affects the cost of product development. Exploratory research is to explore an area where is little known.

In this project, the research is a mixture of descriptive, exploratory and correlational study research. The methodology for design and development knowledge management system is established based on the description of the ‘as-is’ situation of the collaborating company, including organisational structure, business process and users’ requirements. All of these descriptions are the basics of the methodology. This research seeks to define the relationship between knowledge and new product development and figure out why and how knowledge can influence product development. It also helps enterprises to improve the product development through managing product knowledge. This research is mainly exploratory, as many new problems, requirements, processes and relationships are identified during the project, which are not known before hand.

The third perspective is the mode of enquiry used in conducting the study. Research can be divided into two types: quantitative and qualitative. The quantitative research and qualitative research are used for data collection. The qualitative research has un-structured research approach. In other words, the qualitative allows flexibility of the structures. Qualitative research seeks to determine the relationships between a numbers of variables. Interview is a typical method for qualitative research. Quantitative research is structured. Quantitative research seeks to examine relationships within a particular case.

This research is mainly a qualitative type with some quantitative aspects. In the industrial investigation and system implementation, a semi-structured questionnaire interview is adopted as the data collection method. The knowledge involved in this research is complex. All types of knowledge includes figures, models, processes, pictures, reports, meeting minutes and so on will be involved. All of these collected data is used to improve the developed methodology can be used in the real industrial environment. Therefore, this research includes both characteristics of quantitative and qualitative research.

The approach of developing the proposed methodology is shown in Figure 1.1. Firstly, the industrial business process of the collaborating company will be understood, and their requirements are captured. It also can be the investigated problems from the current industries. These requirements are general and implicit, such as a requirement for better manage product knowledge in a manufacturing company. The requirements are investigated through literature survey which tends to find some examples to satisfy the requirements or solve problems. If there is not related research to provide a method to satisfy the requirements from the company, the gaps will be defined, and then a possible solution will be proposed to satisfy the requirements. After the literature survey, the industrial company is investigated, in order to capture data to support the possible solution and enhance it to the proposed methodology. The supporting data of developing the proposed methodology is captured from real industrial environment, in order to ensure that the proposed methodology is suitable for the real industrial requirements. The proposed methodology is verified through a case study in the real industrial environment. Finally, the results of the case study are evaluated in industrial company.

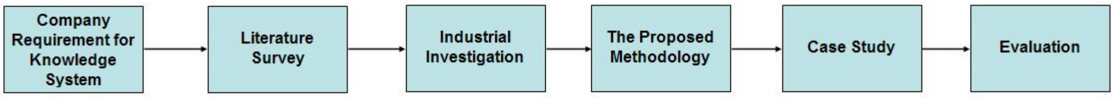


Figure 1.1 – the Research Approach

1.7. Potential Benefit to Industry

The potential benefit of this research is that manufacturing companies may use the developed methodology to produce system specifications of their information systems for the effective management of not only information, but also enterprise knowledge, especially product development knowledge, based on corporate objectives and knowledge users' requirements. Companies could reduce cost, improve efficiency and performance using the KM systems designed by the methodology. The methodology can also be used by information system developers and consultant companies serving the manufacturing industry. The system specifications can be used for two purposes, i.e., (1) for new KM system development and/or implementation; and (2) for assessing companies' existing IT systems in terms of knowledge management capability, and direct the company's future system development and implementation. The developed methodology is based on the Open Group Architecture Framework (TOGAF) which provides a standardised approach to the development and implementation of enterprise wide system design (The Open Group, 2009). TOGAF is driven by requirements of different information users and business objectives. This project will extend TOGAF to include knowledge management capabilities.

1.8. The Collaborating Company

An important criterion of a system development methodology is that the methodology can be actually used in real manufacturing enterprises. Therefore, in this research, the whole developing progress is based on real industrial cases. Currently, many KM research uses manufacturing enterprises as their case studies. Therefore, in this research, an automotive manufacturing enterprise is used as the example. The automotive company is a new company and relies heavily on its advanced new product development process. It is a large manufacturing company, and the total asset of the company is around £1b pounds. The total number of employees is over 20,000. The main products of this enterprise are tractors, dump trucks, platform trucks, buses and sports utility vehicles (SUV).

The current IT situation in the collaborating company is not well managed. It uses a PLM system to help to share product information and cooperation between departments and groups. However, the PLM system is not the only enterprise-level system in the collaborating company. There are several enterprise-level systems to manage different aspects. For example, an Office Automation (OA) system is used for daily work, a Computer Aided Process Planning (CAPP) system is used for manufacturing planning, and a financial information system for budget and cost. These systems are mainly independent of each other. There are also more than 10 department-level systems in use. Some departments even have more than one independent system to support their daily work. There is no communication between department-level systems, and there is no communication between department-level systems and enterprise-level systems. This situation leads to a disorganised structure of information management. The collaborating company needs an entire system strategy to manage all information/ knowledge systematically. More information about the company will be provided in chapter 3.

1.9. Thesis Structure

Chapter 1 – Introduction: This chapter describes the industrial background of the knowledge management system, research domain, aim, objectives, scope and research methodology and approach of this research project.

Chapter 2 – Literature Review: This chapter describes the relevant existing research in the research domain, including knowledge management and its relevant research, requirement management, new product development with the possible modelling techniques in this project, Enterprise Architecture Framework, Product Lifecycle Management and software development.

Chapter 3 – Industrial Investigation: This chapter gives an introduction to the industrial investigation carried out, such as the brief description of the collaborating company, the purpose of industrial investigation, the possible data capture method and the positions of interviewees. This chapter also describes and discusses the results of the industrial investigations, such as the organisational structure of the company, the business process and knowledge user's requirements.

Chapter 4 – The Proposed Methodology: This chapter describes the underlying principles of the proposed requirement driven knowledge management system design methodology to support automotive product development. The methodology consists of a guideline (in the form of a flowchart), an improved Enterprise Architecture Framework based on TOGAF, and a "Folder-based EAF Implementation".

Chapter 5 – Implementation of the Proposed Methodology: In this Chapter, the main structure of the Folder-based EAF implementation is described and discussed, including how to apply the developed methodology to the computer folders. The guideline which system developers can follow will be fully described stage by stage. The links of the Folder-based EAF implementation including the links between the guideline and ADM will be described and discussed.

Chapter 6 – Capturing and Analysing User's Requirements Using the Developed Methodology: This chapter aims to verify and apply the developed methodology in the collaborating company. This chapter describes and discusses how to develop the final list of users' requirements in collaborating company with examples.

Chapter 7 – Developing Knowledge Management System Specifications with the Developed Methodology: In this chapter, the procedure of how to transfer from the final list of users' requirements to the system specification in the collaborating company is described and discussed.

Chapter 8 – Evaluation of the Developed Research Methodology and Discussion of Results: The outcomes of the case study are evaluated in the collaborating company. The outcomes of the developed methodology will compare and contrast with PLM which is current implementing in the collaborating company.

Chapter 9 – Conclusions and Further Work: This final chapter states the conclusions of the research project and explores areas for further research

Chapter 2: Literature Review

The purpose of literature review is to investigate previous research work done in the proposed domain, in order to gather enough knowledge to support proposed ideas. There are 4 key aspects to be investigated: new product development (NPD), knowledge management (KM), requirements management (RM) and enterprise architecture framework (EAF) for formalising the system design. It also required some investigation on supporting technologies for process modelling, software/system development, and Product Lifecycle Management (PLM). In the NPD area, the possible modelling tools to model the involved NPD process and other possible processes will be investigated. PLM is a popular technology to manage the whole product cycle in the current market. PLM is investigated as a comparison of the knowledge system to be designed using the system specifications resulted in this project. Then the concepts of KM will be studied. This part will include the definition of knowledge, the types of knowledge, the methods to transferring the knowledge between individuals, knowledge management process and some examples using knowledge to help organisations. Then requirement management will be investigated. In this project, both product requirements of the collaborating companies and system requirements from the collaborating company are considered. The requirement prioritisation is important to prioritise many requirements in this project. Therefore, requirement prioritisation methods will be investigated, in order to find an appropriate method for this project. One of main part of the literature review is the investigation of the EAF for information system design. It will include introduction and justification of each EAF currently in use, and a comparison will be carried out to support the selection of the most appropriate EAF in this research. Project management is another domain investigated in the literature survey, because most manufacturing companies manage the development of each product as an individual project. Finally, system/software development methodologies are introduced with a focus on a standardised process for developing system specification in this project.

2.1. *Main Issues of New Product Development*

New product development (NPD) becomes one of main efforts in most manufacturing organisations. Usually, a successful product development is determined by 5 factors: good product quality, low product cost, short development time, low development cost, and effective development capability (Kidder, 1981). Therefore, these 5 factors become the objectives in manufacturing business. NPD involves most departments in manufacturing companies. Some departments play main roles in NPD, whilst some others are in supporting roles, such as finance department. In common practice, three departments must be involved (Katzenbach and Douglas, 1993): (i) Marketing department which connects enterprises with customers and captures useful knowledge consisting of customer requirements, market segmentation and product opportunities; (ii) Design department which defines product concepts and designs the final products to meet customer needs; and (iii) Manufacturing department which defines the production planning, scheduling and manufacturing methods, as well as purchasing, distribution, and supply chain management.

In current manufacturing practices, most products are developed as an independent project. Many people cooperate each other to define the product. Ulrich and Eppinger (2011) described a structure of a product development team for an electromechanical product. The team consists of a core team and an extended team. The core team contains team leader, manufacturing engineer, mechanical designer, electronics designer industrial designer, marketing professional and purchasing specialist. The core team identify all the concepts of the product. The extended team includes suppliers to support the core team with the relevant knowledge and materials. Ulrich and Eppinger (2011) defined a generic new product development process which contains six phases (as shown in Figure 2.1).

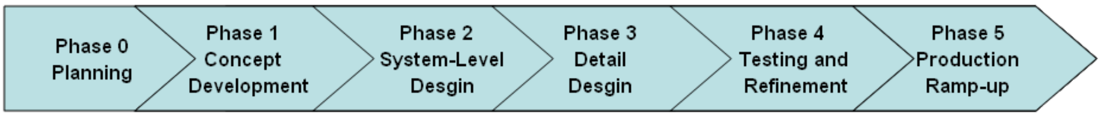


Figure 2.1 – A Generic New Product Development Process (Ulrich and Eppinger, 2011)

Phase 0 is planning, and the purpose of this stage is to identify market objectives and assess the current technologies. The output of this stage is a strategic statement including business goals, missions, key assumptions and constraints. Phase 1 is concept development which is one of most important stages in the process. Product concepts are identified, tested and evaluated in this phase based on customer needs. Phase 2 is system-level design. This phase contains the definition of product architecture and breakdown of the product into subsystems and individual components. The detail design phase (phase 3) contains the complete product specifications, such as geometry, tolerances and materials. In this phase, constraints of the product in implementation are identified, in order to control the risks and failures in actual implementation. Phase 4 is product test. A prototype of the intended product is produced under the constraints and controls. Phase 5 is the production ramp-up. During certain point of this phase, the product will be launched.

This generic product development process is commonly accepted, although there are variations in different manufacturing companies, including the collaborating company of this project. It is noted that the generic process provides a sequential process rather than a iterative process showing feedback or changes.

When a product is developed, it is managed as an individual project, and a project manager is assigned. Meredith and Mantel (2006) defined three objectives of project management, i.e., performance, cost and time. To identify the maximum performance in a limited time period with reliable cost estimation is critical to design management in the manufacturing business. Project management should integrate all aspects in the product development process (Project Management Institute, 1996). Therefore, project management can be used as the basis or starting point for this project to integrate all the stages of new product development within the framework to be developed. Project managers have 3 main responsibilities: plan, organisation and control. In other words, project managers are required to plan, organise and control design projects to finish in time and satisfy all customer requirements (Gido and Clements, 2004). A successful project manager should have many skills, and should be trusted by customers and can motivate members in the project team.

Understanding customer requirements is the starting point in project management. Customer requirements need to be transferred to product design requirements and engineering requirements. Baxter and Gao (2005) reported a methodology to transfer customer requirements to design and engineering requirements. Nilsson and Fagerstrom (2006) developed a methodology to integrate customer requirements with requirements of other stakeholders.

Another factor that can directly influence the success of projects is communication. Good communication can also satisfy the KM requirements in new product development. Shiffler provided a three wise: Communication, Communication, Communication (Project Management Institute, 1998). The formats of communication in project management are multiple, such as oral communication, meetings, telephone calls, emails, letters and internet meetings. The abundant communication provides a good condition of sharing knowledge in new product development. As previous discussed, excellent knowledge sharing is a basic factor of any successful new project development. Therefore, the combination of project management and knowledge management can assure the success of new product development.

There is a common problem with project based product development, i.e., each product development project is carried out independently. The collaboration between projects is limited. This may lead to the continuous product development between generations becoming separated individual projects. This situation could lead to high cost and time waste in the development of similar products. This problem can be improved using the methodologies developed in this research project.

2.2. Modelling Tools in Product Development

The appropriate process modelling tools should be used to plan the project tasks, schedules and resources, and to visually display the details for team members to communicate, share and follow the plan. This section introduces some typical process modelling tools used in product development management, and discusses their advantages and limitations in context of this research project.

2.2.1. Entity-Relationship Model

Entity-Relationship (ER) Model was originally developed by Chen (1976) aiming to represent the relationships between entities. The ER model is widely used in computer database management, because it is good at representing the relationship between different data objects (entities). Figure 2.2 shows a simple example of an ER model. The basic elements are Entity, Attribute and Relationship. In Figure 2.2, the entities are Project and Task. The Relationship is the association between two entities. In the diagram, the Relationship is that the Project consists of Task(s). In this diagram, the Relationship is 1 to 3, i.e., One Project consists of three Tasks. The Attributes represent the particular characteristics of an Entity. In the ER diagram, the Project has a particular name and target costs, which are the Attributes of the Project.

The most important advantage is that the ER diagram represents the relationship very clearly and logically, and its data connection is easy to understand. Thus it is widely used in database creation. Another advantage is that ER diagram uses the English language to describe everything in the diagram, so the structure is easy to understand. The biggest limitation of the ER diagram in manufacturing applications is that the ER diagram does not include any sequence and feedback loop in it. The ER diagram only represents the relationships between entities, it is not enough to model a process based manufacturing design project.

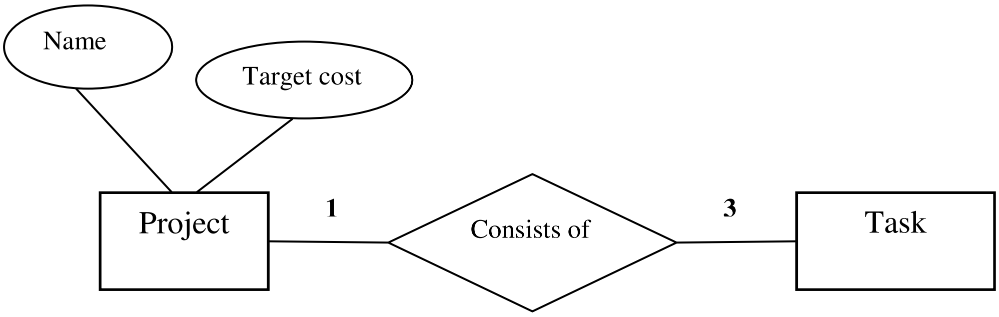


Figure 2.2 – An Entity Relationship Model

2.2.2. Unified Modelling Language (UML)

Unified Modelling Language (UML) is a common standardised modelling language developed by the Object Management Group (Object Management Group, 2011). UML provides a set of notation elements which include classes, components, nodes, activities, user cases, objects, work flow and states, in order to help enterprises specify, visualise, construct and document models of software systems including their structure and design (Sparks, 2000). UML is widely used in different business aspects, such as software development, finance, health, telecom, and aerospace (Fernandez and Moreno, 2004).

UML is widely used in software development and business process modelling in current manufacturing industries. One of the biggest advantages of UML is that UML provides significant and detailed notations and principles for any business sectors (Rosenberg and Scott, 1999), e.g., a use case diagram for visualising the relationships between uses and use cases, a sequence diagram for displaying objects interaction based on time sequences, a collaboration diagram for displaying interactions and relationships between objects, a class diagram for showing relationships between classes, and so on. Moreover, these notations and principles are recognised as a standard by most of researchers and organisations. Therefore, UML can manage most kinds of modelling requirements in current manufacturing industries.

However, UML still has some disadvantages. Firstly, there is a lack of ability for business rule specification. The Object Management Group is still developing UML capable for business rule specification. UML is poor for distributed system design. UML is not capable for sequential development between systems. UML just has capability for simple sequence representation. It is not satisfied for modelling business process, especially for complex and hierarchical processes. UML does not contain any feedback loop for modelling the business process, thus it is not appropriate in this project.

2.2.3. Project Evaluation and Review Technique Model

The Project Evaluation and Review Technique (PERT) model is most often associated with the Critical Path Method (CPM) (Meredith et al, 1985). Actually, PERT diagram and CPM are two different methods for the logical process modelling. The main difference is that CPM is based on the experiences to define the time cost of each activity. The PERT model defines the time cost as a random variable for each activity. The CPM is often used in network planning, and PERT can be used in either networks or organisations.

Today, most organisations use PERT in association with CPM, because CPM can be used to save time cost and improve efficiency as main purpose, whilst PERT can point out the key reasons of saving time and improving efficiency. If they are integrated together, organisations can gain the advantages from both of them.

Normally, a PERT diagram represents a sequence of the project activities, and the user can find the critical path in the diagram. Building a PERT diagram needs to be based on a list of logical activities. Each activity should have the time cost and some constraints of the sequence. A typical PERT diagram is shown as Figure 2.3. The sequence is: start from F, and then E, A, B, C. The length of an arrowed line represents the duration of the transition. Sometimes, the duration can be marked on the arrowed line in the diagram.

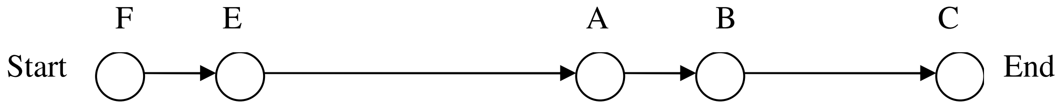


Figure 2.3 – An Example PERT Diagram

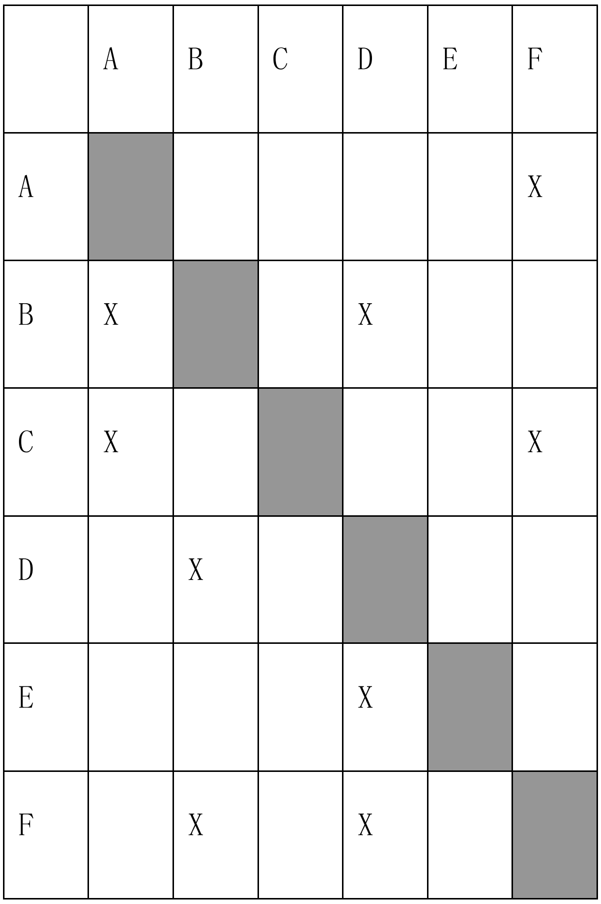
PERT is a standard representation for project management, because it is most useful for representing the time which is required to achieve milestones, and it can define the critical path (Park and Cutkosky, 1999). However, PERT cannot represent the circuits, so feedback loop is not represented in PERT. In manufacturing industry, the design process model often needs to be included feedback loops, because many requirements need to be compromised with constraints and thus changes have to be made in decisions taken earlier. Another limitation of PERT is that a large PERT diagram is very difficult to maintain when frequent changes are required (Steward, 1981b).

2.2.4. Design Structure Matrix

Design Structure Matrix (DSM) uses a matrix representation to overcome the size limitations of other process models. It is developed by Steward (1981). As a system analysis tool, it provides two dimensions to capture the dependences between organizational elements (Woodman and Bilardo, 2005).

DSM can represent large number of relationships between elements. A simple DSM is shown in Figure 2.4 which represents the dependences of six elements. The "X" represents the dependence relationship between two elements. For example, element A determines element F (read from left of the matrix). Element A is determined by B and C (read from top of the matrix).

The DSM model is good at finding the dependence when the process model includes complex and large number of relationships, and the sequences are less important in the project (or the sequences are represented in another model), and the DSM is easy to be built up (Keith and Vincent, 2005). Microsoft Excel can be used to produce the matrix easily (Woodman and Bilardo, 2005). However, DSM also has its limitations. Park and Cutkosky (1999) pointed out that a series of dependencies between connected nodes in a process model is not immediately apparent, and this makes redundant and parallel paths difficult to distinguish. It also has limitation on the process sequence. If the sequence is complex and very important, DSM is hard to deal with this problem.



	A	B	C	D	E	F
A	▨					X
B	X	▨		X		
C	X		▨			X
D		X		▨		
E				X	▨	
F		X		X		▨

Figure 2.4 – An Example Design Structure Matrix

2.2.5. Integrated Computer Aided Manufacturing (ICAM) Definition Modelling Technique

IDEF (Integrated Computer Aided Manufacturing (ICAM) Definition) modelling technique is used to model decision-making processes including activities and actions in an organisation or a system. IDEF is derived from the Structure Analysis and Design Technique (SADT) which was developed by Ross of the United States Air Force (Ross, 1977). IDEF is usually used to analyse the complex processes, and also used in functional analysis for a system and to identify which functions are performed, why these functions are needed. IDEF models are usually established as one of the first tasks of a system development project (IDEF Web site, 2007).

An IDEF model uses arrowed lines and boxes to show the functions and represent activities. Figure 2.5 represents the basic syntax of an IDEF0 diagram which is the basic diagram in IDEF family. IDEF0 diagram represents the conditions of an activity distinctly. The left and right arrowed lines represent the inputs and outputs of the activity, and the top arrowed lines are the constraints or triggers for the activity. The bottom arrowed lines represent the mechanism of the activity.

All members of IDEF family have strict rules to make constraints for the diagram. For example, IDEF cannot include more than six activities in each hierarchy. Thus, it may lead to large and complex IDEF diagrams when an organisation is large and its business is complex. IDEF0 has several advantages because of the strict rules. The first advantage is that is has a good elasticity and logical structure. It can help organisations manage the detailed perspective of the whole structure of a project. Another advantage is that organisations can divide the project into details, and everyone involved can understand his/her work clearly. The third advantage is that IDEF0 can be integrated with other IDEF family methods. Thus, IDEF0 is particular appropriate for large projects.

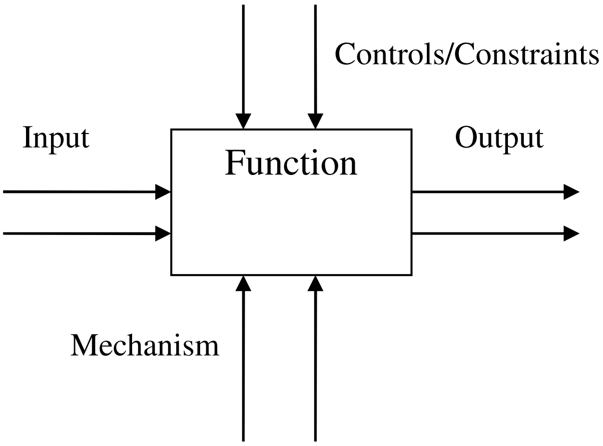


Figure 2.5 – the Basic Syntax of IDEF0 Diagram

IDEF0 also has some limitations. The first limitation is that the structure can be very complex for large organisations. Thus, IDEF0 models need experts to create. And may involve many people in the whole project, thus the cost may be high. Another limitation is that the activity sequences are not very clear in IDEF0, thus continuous activities are difficult to understand and model. Although feedback loops are included, they are difficult to trace when a model crosses more than one hierarchy.

2.2.6. Design Roadmap

Design Roadmap (DR) was developed by Park and Cutkosky (1999). The original purpose was to seek a method to overcome the limitations of process representations discussed above. Park and Cutkosky developed this technique to provide a comprehensive method for project management. The basic elements of a DR model are Tasks and Features. Tasks are the primary elements of the process model. Features are the input and output of Tasks. Thus every Task has a Feature as its input, and another Feature as its output. The arrowed lines are used to represent the process flow and links between the Tasks and Features.

A DR model also has complex dependencies. In these dependences, the feedback dependency is most often used. The feedback loop is needed in the design process. For example, when the engineering requirements need to be integrated with customer requirements, the engineers will need to discuss with the sales and marketing people to see whether customer requirements can be modified. A feedback loop is needed between the output of the engineering requirements and the customer requirements.

Figure 2.6 shows a simple DR model. Feature A is the input of Task 1, and Feature B is the output of Task1. Similarly, Feature B and C are the input and output of Task 2 respectively. There is a feedback loop between Feature C and Task 1, i.e., the result of Task 2 (Feature C) is considered by Task 1 which may result in changes in Feature B. The DR model enables sub-systems (i.e., sub-models) to be contained in Tasks and Features.

DR models can deal with both simple processes and complex processes. The syntax of DR is easy to understand and build. DR is particular appropriate for manufacturing projects, because it is good at representing sequences and feedback loops. DR normally does not require a particular system programme to produce it, and

Microsoft Excel can produce a perfect DR model. However, DR is a not yet a commonly used method such as IDEF which is regarded as an international standard.

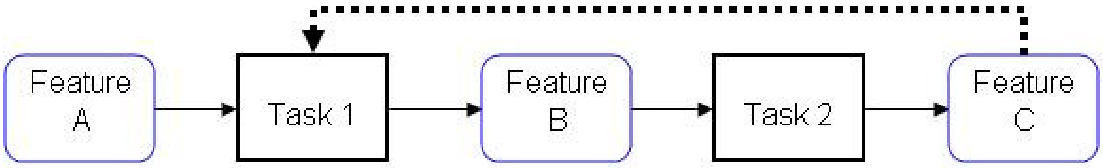


Figure 2.6 – An Example Design Roadmap Model

2.3. Product Lifecycle Management

Product Lifecycle management (PLM) is a process of managing the whole lifecycle of a product consisting of its conception, design, manufacture, service and maintenance (CIMdata, 2009). PLM systems integrate people, data, processes and business systems and provide product information backbone for companies and their extended enterprise (PLM Technology Guide, 2009). PLM tasks include the creation, mortification and exchange of product information (Srinivasan, 2008).

PLM is accepted as one of the most popular tools to manage product development. Many manufacturing companies invested millions of pounds in research and development of this area (O'Marah and Myer, 2002). Many researchers contributed to this subject. The earlier researchers were Abernathy and Utterback (1975, 1978). They defined the general concepts of PLM and also defined a dynamic model for product innovation and its process. This model was one of the earliest models, and it also was the initial model for PLM system development.

Many international companies developed systems and software packages for PLM. Sudarsan et al (2005) summarised the domain of the applications for their resulting PLM system with a common PLM architecture as shown in Figure 2.7. The architecture consists of an Information Technology (IT) infrastructure; a product information modelling architecture including ontology and inter-operability standard; a development toolkit with its environment to provide the means for building the business applications; and a set of business applications which provides PLM functionality for processing the corporate intellectual capital.

Sudarsan et al (2005) developed a product information modelling framework which is based on the National Institute of Standards and Technology (NIST) core product model and its extensions. This information modelling framework can support the full range of the PLM information requirements. The framework is intended to capture, design and assemble product information in the full product lifecycle. The framework eases the inter-operability of the Computer-Aided Design and Manufacturing (CAD/CAM). One of the main characteristics of the framework is to capture the evolution of products and its generations. In other words, it emphasises the reuse of product knowledge in further product design. There are some difficulties of this framework. One obvious difficulty is the whole framework is too large to find the appropriate information when it is necessary. Hence an information search methodology is required. The framework can not be finished with one single language, particular in the application programming interface (API). It may lead to a large implementation to finish the framework.

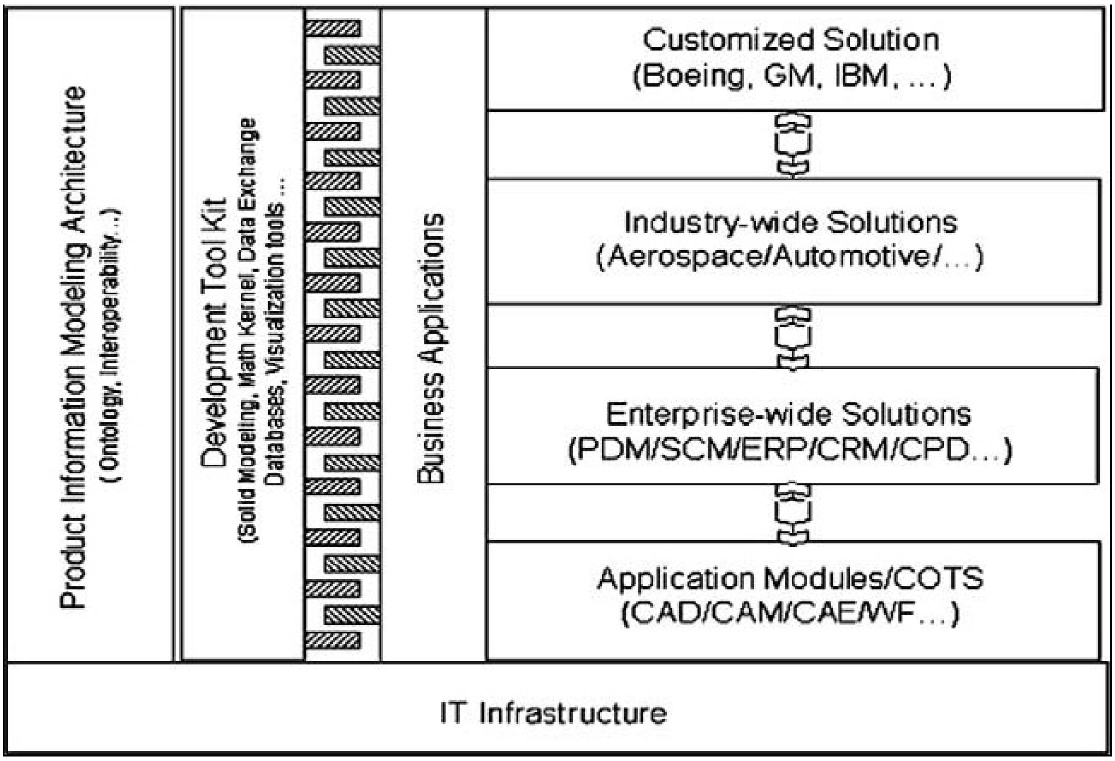

Product Information Modeling Architecture
(Ontology, Interoperability...)
Development Tool Kit
(Solid Modeling, Math Kernel, Data Exchange
Databases, Visualization tools ...
Business Applications
Customized Solution
(Boeing, GM, IBM, ...)
Industry-wide Solutions
(Aerospace/Automotive/...)
Enterprise-wide Solutions
(PDM/SCM/ERP/CRM/CPD...)
Application Modules/COTS
(CAD/CAM/CAE/WF...)
IT Infrastructure

Figure 2.7 – A Conceptual PLM System Architecture (Sudarsan et al, 2005)

Srinivasan (2008) suggested that the current Information Technology (IT) tools should have the capability for wider collaboration and integration to achieve the business objectives. However, the current IT system cannot meet the challenges of sharing information in a complex system with a flexible method. Srinivasan identified 3 important developments that should be changed in IT tools: (1) data and meta-data models, engineering and business processes are standardised and maturity; (2) service-oriented architecture (SOA) for information sharing; and (3) an appropriate robust middleware to implement. Srinivasan developed the integrated framework based on business requirement and these 3 developments. The obvious advantage of the framework is that all the executions of the framework are based on the business needs. It can guarantee the developed framework would meet business requirements.

Currently, many manufacturing companies use PLM systems to help achieve good product quality, low product cost, short development time, low development cost, and effective development capability, and solve the problems existing in NPD. One of the global leaders in PLM called CIMData (2008) defined PLM as a process to manage the whole product lifecycle from the product concept form to the service for the customer. PLM not only solves the problems and manages the complete product approach, but also manages all the data and information in this approach (PLM Technology Guide, 2008).

Abramovici and Soeg (2002) identified several benefits of PLM. PLM helps manufacturing companies reduce product time to market. It can improve product quality and reduce prototyping costs. PLM provides a framework for product optimisation, in order to reduce waste and save through the complete integration of engineering workflows.

There are still many research work done recently, such as Ming et al (2007) suggested a collaborative process planning methodology for PLM. Tang and Yun (2007) developed a data model for quality in PLM. However, PLM systems have limitations in managing enterprise knowledge. The flexibility of PLM system can be further improved. PLM system implementations should be driven by business objectives and end users' requirements, but in real life this is not always the case. The current high-end PLM systems or methodologies are costly for implementation and maintenance. It is not always suitable for small and medium enterprises.

2.4. Knowledge Management in Manufacturing

2.4.1. Definition of Knowledge

Manufacturing operations become more knowledge intensive and companies are investing significant resources in Knowledge Management (KM) (Choi et al, 2008). KM is identified as a necessary business function for business process improvement and for maintaining competitive capability when enterprises lost their key personnel (Briggs, 2006). It influences cost, time and successful rate of the development directly.

Before defining knowledge, it is useful to clarify the related terminologies: data and information. Data and information are often confused with knowledge. Wilson (1987) defined data as the representation of information independent of meaning. Data can exist independently; can represent single meaning, e.g., 96 cm. Information can be viewed as the data plus the meaning connected to it, e.g., the identified diameter of the car wheel is 96 cm. Toften and Olsen (2003) also defined information as a set of data collection including management, marketing, organisational behaviour and social policy decision making. From KM perspective, Nonaka (1994) defined information as flow of messages.

Knowledge is a familiar concept for everybody, but the definition of knowledge is not so straight forward, because it is a non-objective concept. An authoritative definition of knowledge was given by the European Committee for Standardisation (2004): knowledge is the combination of data and information, to which is added expert opinion, skills and experience, to result in a valuable asset which can be used to aid decision making. Within the definition, knowledge can be understood simply as a set of data, information, expert opinion, skills and experience for decision making. An example can be given: the identified regular diameter of the car wheel is 96 cm; the car wheels with diameters 92cm, 94cm, 98cm, 100cm can be fitted in the same car body.

According to several researchers, knowledge is something more than information, and it can only exist in the individual mind (Alavi and Leidner, 1999, Court, 1997, and Rodgers and Clarkson, 1998b). Davenport and Prusak (1998) provided a definition which is widely cited in literatures: "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 knower. In organisations, it often becomes embedded not only in documents or repositories but also in organisational routines, processes, practices, and norms".

2.4.2. Types of Knowledge

Wyatt (2001) indicated that knowledge can be classified into explicit and tacit. Explicit knowledge can be expressed in formal methods or natural languages. It can be shared and exchanged as formal data, formulae, or documents. Tacit knowledge is normally not expressed or even cannot be expressed, emphasising personal skills and understanding, and is very difficult to share and exchange by formal and systematic methods. This kind of knowledge is the opinion, experience and action based skills (Miller, 1998). Therefore, how to manage both explicit and tacit knowledge becomes a real problem in practice.

Manufacturing involves both kinds of knowledge, and both of them are important in industry. Thus, the conversion and sharing of knowledge are vital in the current industrial practice. Normally, there are four conversions and sharing of knowledge. The first one is to convert the tacit knowledge from one person to another person's tacit knowledge. This conversion is a social interaction of people to share their knowledge each other. Nonaka and Teece (2001) explained tacit knowledge accumulation in detail for sharing tacit knowledge in organisations. The second conversion is to transfer tacit knowledge to explicit knowledge. This process is very difficult, but sometimes it is necessary such as when many people need to learn some skills at the same time. The next conversion is from explicit to explicit, and this kind of conversion happens when two explicit knowledge need to be integrated. For example, two design process models need to be integrated in order to develop a new process model with a formal language. The last conversion is to exchange the explicit knowledge to tacit knowledge. This conversion usually happens in organisations when people need to understand a formal presentation, such as the handouts and specifications, and then this formal presentation is understood and learnt by individuals, and becomes his/her internal knowledge.

2.4.3. Concepts of Knowledge Management

A good method of managing knowledge is an important issue in new product development. Many researchers define KM in their own opinions. Drucker (1998) defined KM as the ability to harness and build upon an organization's intellectual capital. This definition was one of the earliest definitions of KM. European Committee for Standardization (2004) defined KM as the management of activities and processes for leveraging knowledge to enhance competitiveness through better use and creation of individual and collective knowledge resources.

KM can help organisations to gain advantages in competitiveness and create more values for organisations. First of all, organisations need to know where the knowledge exists. In organisations, some knowledge can be found directly as the capital of the organisations, such as the literature, database, annual reports and financial reports. All the people can access them with the appropriate permission. Saint-Onge (1996) defined knowledge as intellectual capital in organisations. The intellectual capital contains the human capital which is the knowledge of each individual staff, the customer capital which is value perception of customers and organisational capital which is the knowledge that has been captured within a structure, such as a process model to produce a product.

KM can help organisations both in long term and short term (Sarvary, 1999). In the short term, KM can help organisations to model a particular project process and capture the intellectual capital in organisations. In the long term, KM can improve the business performance of organisations, and it is incoporated with the basic purpose of each organisation. From the historical experience, if an organisation can manage intellectual issue, such as human knowledge, the organisation would be successful easily. Better KM can improve the reuse of existing knowledge in organisations (Zdrahal et al, 2007), consequently organisations will not waste their valuable knowledge that had been obtained through previous expensive investment.

Knowledge Management consists of several activities, such as knowledge acquisition, knowledge storing, knowledge sharing, knowledge use/reuse, and knowledge integration. First, existing knowledge needs to be captured. The methods of capturing knowledge can be formal and informal. The formal methods are interviews, workshops, experiments and instruction. The informal methods are more like social ways, such as communications between engineered managers.

Ratchev et al (2002) developed a model named KARE (Knowledge Acquisition and Sharing for Requirement Management), in order to provide a structured method for acquiring relevant enterprise knowledge. KARE also provides an interface for the system requirement review. KARE contains two parts in it, one is the requirement engineering, and the other is knowledge engineering. The requirement engineering part contains a three steps process – requirement elicitation, analysis and negotiation. Knowledge engineering part contains elicitation and analysis. This model supports engineers to capture and manage knowledge based on requirements.

After knowledge is captured, it needs to be stored. Knowledge can be stored in many different ways. It can be stored on paper by hand-writing. It can also be stored in human memory. It can be stored in databases. However, to find the best way to store knowledge is still a pending problem, and how to store tacit knowledge like human experiences is a difficult problem. Actually, tacit knowledge should be expressed first, and then it may be stored. Otherwise tacit knowledge remains with the knowledge owners, and a meta data about 'who has what knowledge' can be stored in the organisation's knowledge base.

Knowledge can be used/re-used after storing, and can also be shared among many people who collaborate in a project in enterprises (Gold et al, 2001). Then there should be some new knowledge created during the project. Then the new knowledge becomes existing knowledge for future projects.

2.4.4. Current Research in Knowledge Management

Many organisations use knowledge management for specific problems solving (Vestal, 2005). Therefore, some researchers suggested a comprehensive way to manage knowledge systematically in organisations. Knowledge based engineering system is one of the solutions. Knowledge based engineering (KBE) is a technique to help organisations to transfer and maintain useful knowledge in engineering process. KBE is defined as an advanced software techniques to acquire and re-use knowledge in product development process in an integrated way (Stokes, 2001). Therefore, knowledge acquisition, transferring and modelling are very important for new product design and innovation (Kalpic and Bernus, 2002).

A KBE methodology was developed by a Consortium called Methodology for Knowledge Based Engineering Applications (MOKA). MOKA provided a methodology containing the principles of KBE, a process to achieve the models and software to support it (Stokes, 2001). MOKA described a 6 steps lifecycle of KBE consisting of Identify, Justify, Capture, Formalise, Package and Activate, as shown in Figure 2.8. The purpose of the Identify step is to investigate the business requirements and determine the types of the KBE system which can satisfy these requirements. It is also the trigger of the next step, thus all of the values need to be identified clearly in this step. If not, some uncertain values will be sent back and should be defined again.

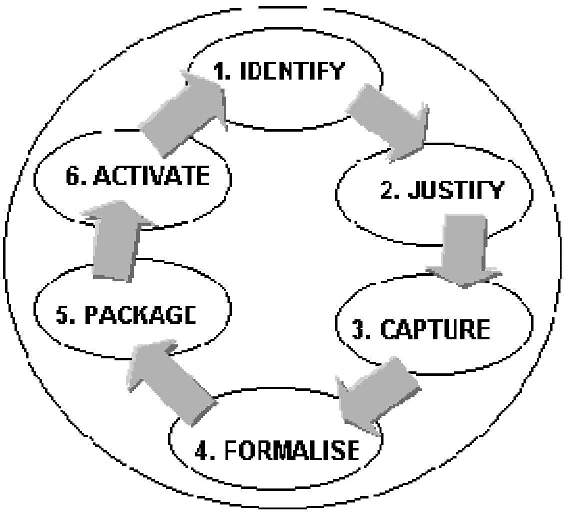


Figure 2.8 – MOKA KBE Lifecycle (Stokes, 2001)

The Justify step aims to manage aspects of risk assessment, cost estimation and timing. Also, this step determines the conditions of a successful project. After the Justify step, the next step is the Capture step, and the main purpose of this step is to capture the knowledge in the engineering process, such as to extract knowledge from a design expertise. After that, the knowledge can be Formalised and packaged in databases or knowledge warehouses. Finally, this knowledge can be activated by users. MOKA also developed models and systems based on this lifecycle, thus MOKA provided a successful case study of knowledge based engineering.

However, MOKA is not finished as a commercial system. Although it is not a completed commercial system, it provides a good idea for KBE system. Some other researchers tried to solve the same problems from another angles. Many researchers claimed that an entire knowledge management system can help organisations to manage their knowledge systematically.

Chua (2004) suggested that a comprehensive KM system should consider the fundamental KM processes such as knowledge acquisition, transferring, storing and sharing. Chua (2004) defined KM system architecture with 3 layers, i.e., the presentation layer with personalisation and visualisation functions; knowledge services layer with knowledge creation, sharing and reuse functions; and the infrastructure services layer with knowledge storage and communication functions. He suggested that this representation is suitable for social, economic and technological factors of an enterprise.

Jung et al (2006) reported an integrated architecture for knowledge management systems and business process management systems based on process-oriented knowledge management, in order to combine the advantages of the two paradigms (Berztiss, 2000, Maurer and Dellen, 1998, Maurer and Holz 1999, Choi and et al., 2004, Lai and Fan, 2002). In the initial stages, the knowledge management systems and business process management systems lifecycles are integrated. These two lifecycles have similar structure and steps which is shown as Figure 2.9. This integration can help enterprises set up their business process based on the knowledge structure, and also enterprises do not need to construct another knowledge structure separately when the knowledge management is necessary to be contained.

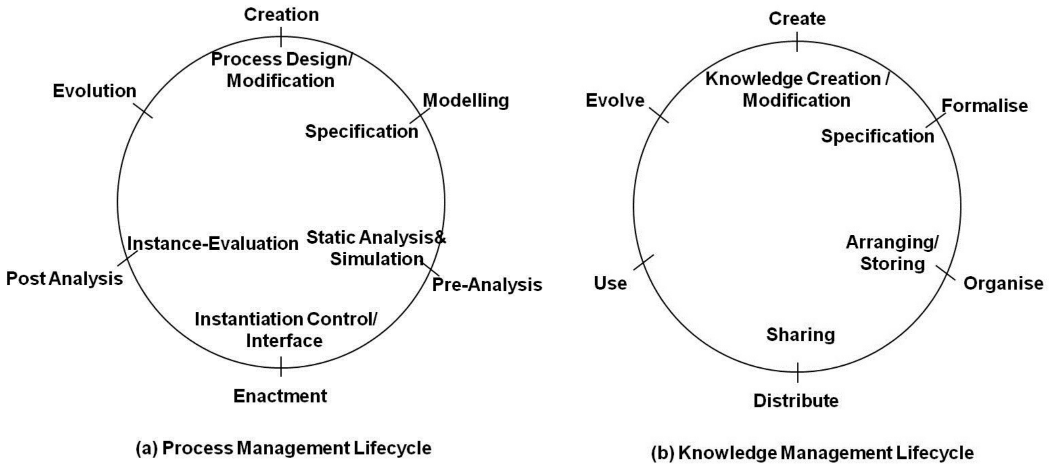

Creation
Process Design/ Modification
Evolution
Modelling
Specification
Instance-Evaluation
Static Analysis& Simulation
Post Analysis
Pre-Analysis
Instantiation Control/ Interface
Enactment
(a) Process Management Lifecycle
Create
Knowledge Creation / Modification
Evolve
Formalise
Specification
Arranging/ Storing
Use
Organise
Sharing
Distribute
(b) Knowledge Management Lifecycle

Figure 2.9 – BPM lifecycle and KM lifecycle (Jung and et al., 2006)

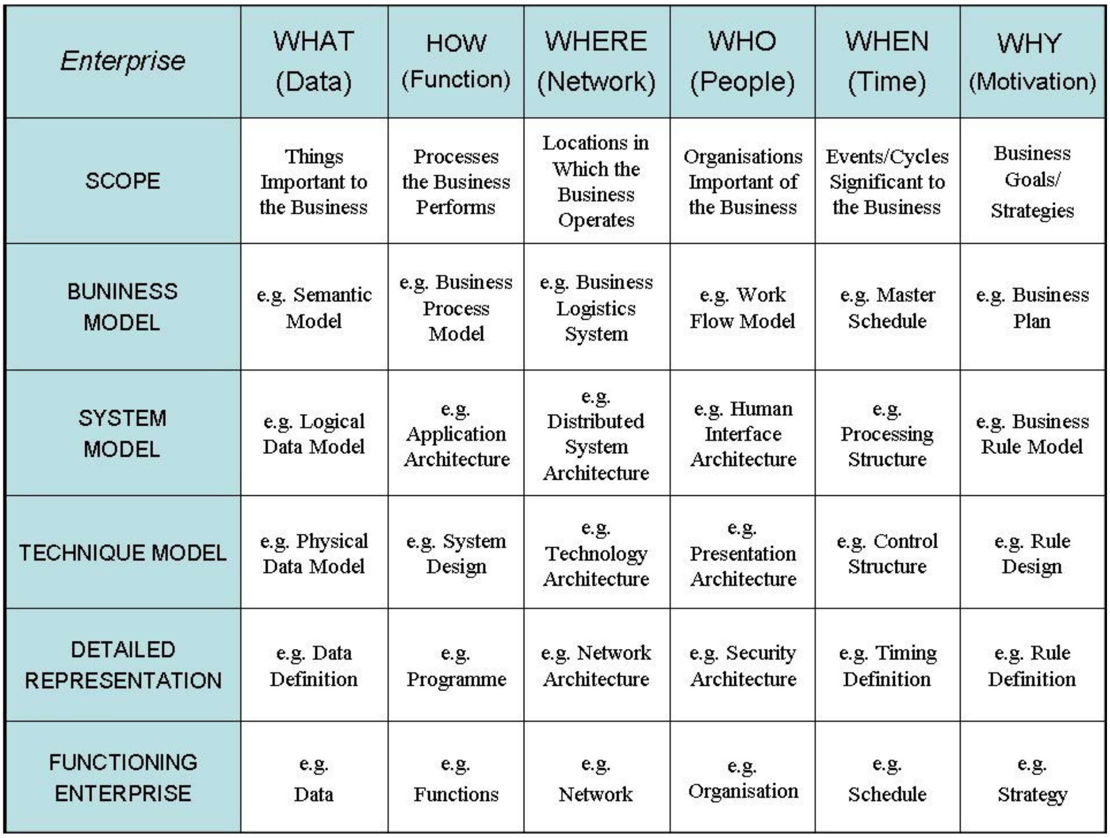
This integration can help enterprises develop their knowledge structure based on in-house business processes. They also constructed the interrelationship between the two lifecycle processes so that they cooperate with each other. This research provided a prototype process-oriented knowledge management system. However, just combining knowledge management lifecycle and business process management lifecycle is not enough for building up an enterprise-level knowledge management system. However, system users' requirements were not considered in this research.

Information Technology systems have been more widely and successfully implemented in business, compared with the less mature KM systems (Burrows et al, 2005). Chen et al (2003) suggested that traditional IT systems are not designed to support organisation learning. They are normally passive and straightforward. Information is only provided when users ask for it. They do not provide proactive communication in an enterprise. They normally provide a single port of accessing databases, and this does not satisfy the real working environments of most enterprises. In the current industrial circumstances, a desired system should be like a workshop which can manage communication, working relationship, people behaviours, knowledge idea, culture and context of enterprise (Ruggles, 1998, Bush and Tiwana, 2005).

In summary, current information systems can not satisfy industrial requirements for better managing enterprise knowledge. There are no formal methodologies to follow when enterprises are planning, designing and implementing KM systems. Attempts to finding a satisfying technology have been made by researchers and practitioners. A formal framework which can be used to guide the design and implementation of knowledge management systems may be proposed on the basis of the widely used Enterprise Architecture Frameworks (EAFs), which are introduced shortly.

2.5. Enterprise Architecture Frameworks

Enterprise Architecture Frameworks (EAFs) are developed to help address the complex systems and business alignment problems in information system design and implementation (Sessions, 2007). The earliest and most popular EAF is Zackman Framework which consists of two dimensions described as perspectives and abstractions of an enterprise (as shown in Figure 2.10) (Zackman, 2011).



Enterprise	WHAT (Data)	HOW (Function)	WHERE (Network)	WHO (People)	WHEN (Time)	WHY (Motivation)
SCOPE	Things Important to the Business	Processes the Business Performs	Locations in Which the Business Operates	Organisations Important of the Business	Events/Cycles Significant to the Business	Business Goals/ Strategies
BUNINESS MODEL	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Work Flow Model	e.g. Master Schedule	e.g. Business Plan
SYSTEM MODEL	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Processing Structure	e.g. Business Rule Model
TECHNIQUE MODEL	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
DETAILED REPRESENTATION	e.g. Data Definition	e.g. Programme	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Definition
FUNCTIONING ENTERPRISE	e.g. Data	e.g. Functions	e.g. Network	e.g. Organisation	e.g. Schedule	e.g. Strategy

Figure 2.10 – Zackman Framework (Zackman, 2011)

Zackman Framework contains six columns, six rows, totally 36 cells in the framework. The columns respectively represent what (data), how (function), where (network), who (people), when (time) and why (motivation). The six rows represent scope, business model, system model, technique model, detailed representation and functioning enterprise. Many enterprise managers consider Zackman Framework as the combination of enterprise rules and IT development.

This framework consists of an enterprise architecture and an information architecture. The framework complies with many kernels of other methodologies, and aims to build up an independent platform for enterprise system development (Noran, 2003). However, Zackman Framework has several weaknesses which are not suitable for this project. It is a static framework involving heavy documentation (Ambler et al, 2005). Therefore it might not be suitable for application in the dynamic new product development process, e.g., it doesn't represent business processes.

The US Department of Defense developed an architecture framework, called DoDAF (Department of Defense Architecture Framework Working Group, 2003), which used three views instead of the six abstracts used in Zackman Framework. The three views are: operational view, system view and technical standard view. The Core Architecture Dada Model and architecture products are required to prescribe the documentation of the architecture with these three views. The data of three views are defined in the Core Architecture Data Model including an overview, summary and integrated dictionary of the architecture (Tang et al, 2004).

Compared with other EAFs, DoDAF does not provide a holistic approach to the design of information systems (Eden and Kazman, 2003). Because the framework is specifically designed to support defence operations, processes and taxonomies, it does not provide business principles such as business plan and investment plan that are the basic requirements of product development.

Another available EAF is the Federal Enterprise Architecture Framework (FEAF) (CIO Council, 2001) which provides an organised structure and a comprehensive classification of Federal Agencies in complex federal structures. The advantage of the FEAF is that it contains two segments: core mission area segment and business services segment (as shown in Figure 2.11). Obviously, when this framework is designed, the use of the framework is designed for both government and commercial organisations.

One characteristic of the reference models of the FEAF is that all of the references are established by the common languages in order to communicate, cooperate and collaborate through the political boundaries (FEA Consolidated Reference Model Document V2.1, 2006). There are 4 main steps of the FEAF. Step 1 is the architecture analysis, and this step is to define a simple for the segment and relate it back to the organisational plan. Step 2 is the architectural definition. In this step, all the states in the architecture are identified such as the relevant business, data, services and appropriate technology which are necessary in the architecture. Step 3 defines the cost and budget to develop the architecture as one individual project. Step 4 creates a management plan and executes the project.

Compared with other Frameworks, this framework provides an architecture process, and can be viewed as an implementation methodology for the creation of enterprise architecture. However, FEAF does not provide development methods for each cell in the matrix structure (Medvidovic and Taylor, 2000).

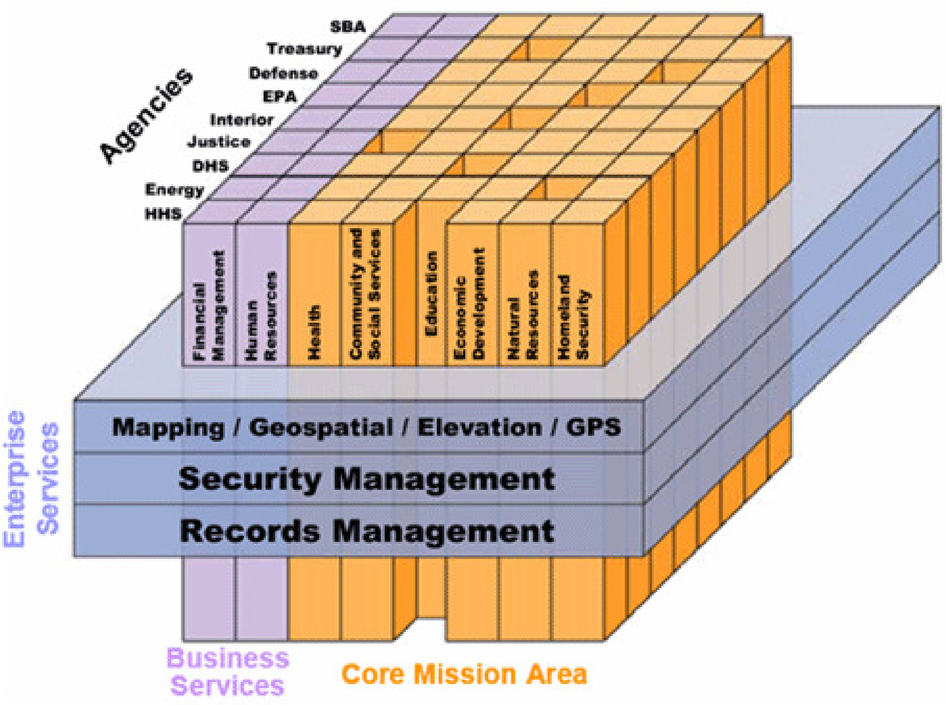


Figure 2.11 – Federal Enterprise Architecture Framework (the CIO Council, 2001)

As one of the most famous IT research and consulting institutes, Gartner suggested an enterprise architecture framework in 2005. Gartner (2009) regarded the enterprise architecture as the process of translating business and strategy into effective enterprise change. The main purpose of the Gartner Framework is to build up an enterprise architecture practice.

Gartner Enterprise Architecture Framework (GEAF) contains 3 primary viewpoints: Enterprise Business Architecture (EBA), Enterprise Information Architecture (EIA) and Enterprise Technology Architecture (ETA). EBA describes the process of business process and organisational concerns, EIA represents information flow and information modelling architecture, ETA concerns the fundamental technical implementation and operations (James et al, 2005).

Gartner framework is specially focuses on the business. The further development is strong and enough. The model is maturity, and especially the practice guidance is the best of all of EAFs. However, the information of the framework is not organised very well. The reference models are weak comparing with other EAFs.

The Open Group Architecture Framework (TOGAF) (The Open Group, 2009) aims to speed up and simplify the development of enterprise information systems. TOGAF creates 4 different architectures for enterprises, i.e., business architecture, data architecture, application architecture and technology architecture. It can help enterprises avoid failure and unnecessary cost when architecture framework is developed. Enterprises can easily select any of the 4 architectures for their particular requirements.

TOGAF provides a 3 layers structure consisting of an Architecture Development Methodology (ADM), an Enterprise Continuum and a Resource Base (the Open Group, 2007). These three layers cooperate and support each other. ADM provides an architectural process to help enterprises to build up EAFs based on their special requirements. Enterprise Continuum reflects different levels of abstraction in a development process, and provides a context with continued construction of the EAF. The Resource Base provides technical support on data management, in order to apply ADM. ADM establishes requirements as the centre of an architecture framework.

These requirements are information system requirements based on each individual group or activity. The Architectural process of ADM is shown as Figure 2.12.

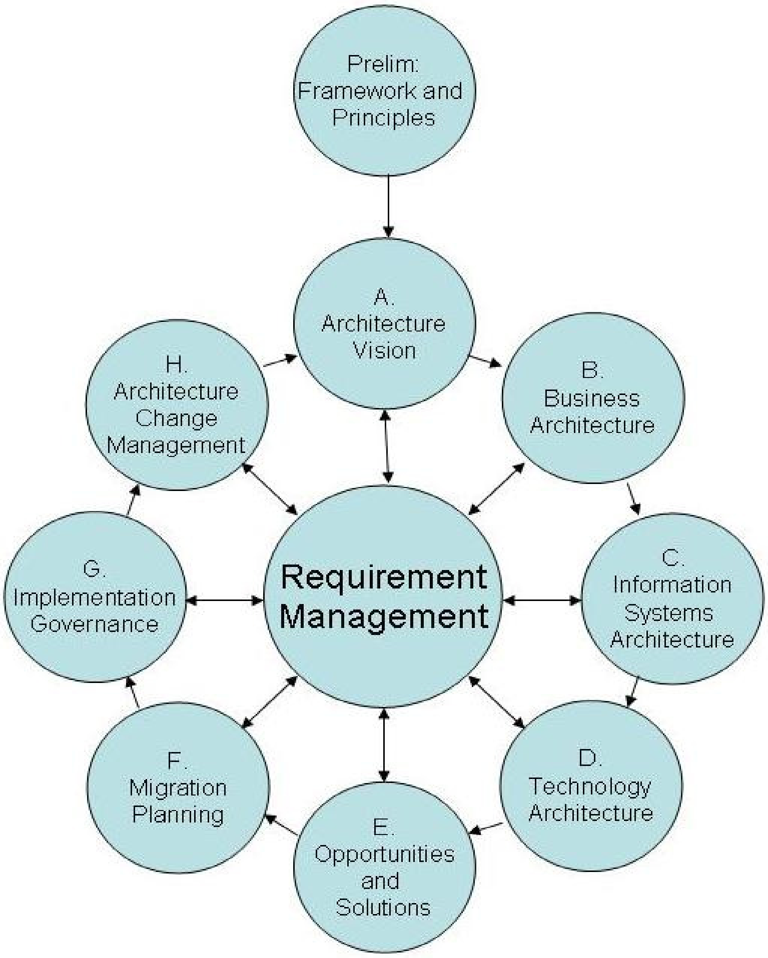

Prelim: Framework and Principles
A. Architecture Vision
B. Business Architecture
C. Information Systems Architecture
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirement Management

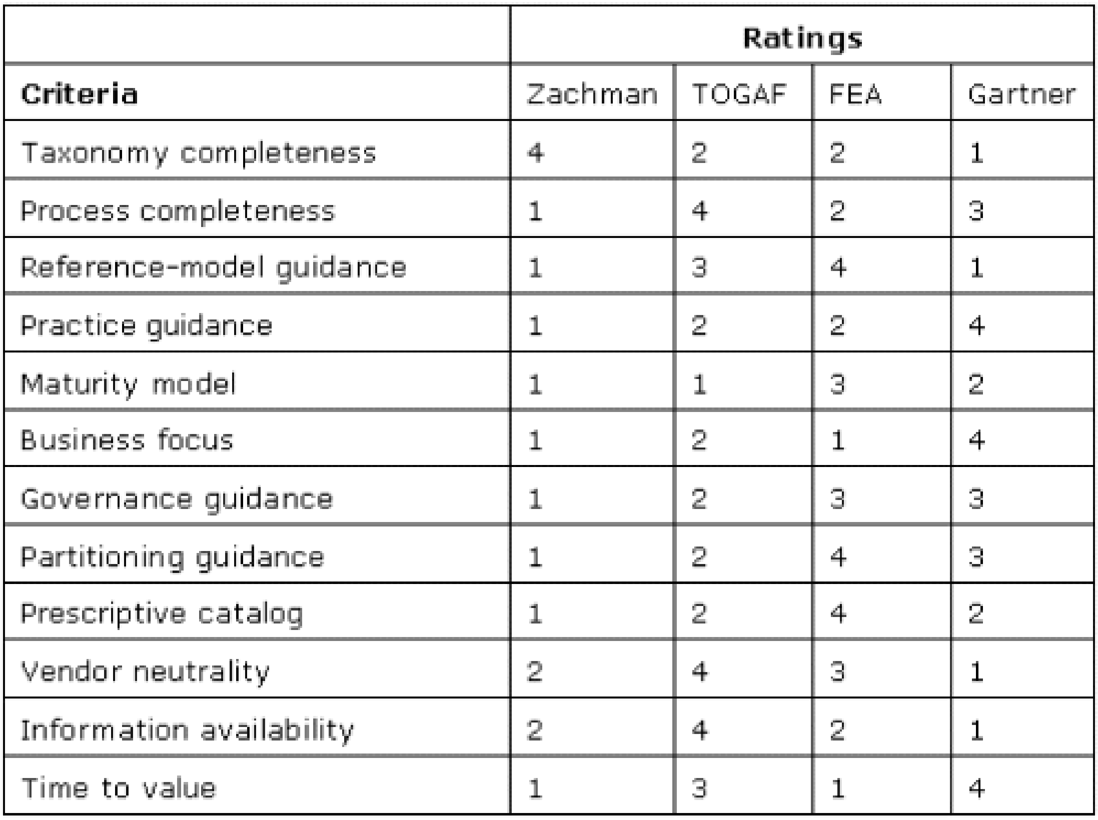
Figure 2.12 – ADM of TOGAF (The Open Group, 2008)

The first stage is the preliminary stage. The main objective of this stage is to define the roles that should be involved in. This approach is committed to the success of the architectural process. The next step (A) is the architecture vision. This step is used to identify the business principles, business goals, and strategic business drivers of the organisation. A formal development process will also be determined in this step. In the next stage (B), the business architecture will be defined. In this step, the business architecture can be a process or a structure of the actual business. After the business architecture is ready, information architecture will be developed (Step C) based on the business architecture, it contains two parts: application architecture and data architecture. The next step (D) will develop the technology architecture. And then, the possible opportunities and solutions will be worked out in Step (E). These opportunities and solutions will be prioritised based on several difference constraints such as costs and benefits in the next Step (F), and also a detailed implementation plan is developed in this step. The implementation governance Step (G) will formulate recommendations for each implementation plan which is developed in the previous step. And the final Step (H) is to establish an architecture change management process for the new enterprise architecture baseline. All the main steps are linked to and driven by the user requirements as shown in the centre of the figure.

TOGAF is more like a process to guide the development of an EAF rather than a framework structure (Sessions, 2007). Therefore, TOGAF can help enterprises build their own EAF with their own processes, whilst Zackman Framework does not provide a process to guide EAF development. However, knowledge management and product development have not been addressed as main themes in the development of TOGAF. This is what this research project will focus on.

As described above, the existing EAFs in the current market have different focuses. Many researchers have given their comments on the existing EAFs and analyse them for different situations. Tang et al (2004) suggested an analysis with the characteristics of the fundamental elements such as goals, inputs and outputs. Sessions (2007) produced a fully comparison of the top 4 enterprise architecture frameworks as shown in Figure 2.13. Sessions produced 12 criteria for the 4 top EAFs with the rating from 1 to 4. "1" represents "very poor job" in the criteria. "2" represents an "inadequate job" in the criteria. "3" represents "acceptable job". "4" represents a "very good job".

From the rating and criteria, the characteristics of each framework are very clear. Zackman Framework has a good Taxonomy Completeness, and none of the other frameworks focus on this area. However, Zackman Framework is weak in other criteria. TOGAF has a good performance in the Process Completeness, Vendor Neutrality and information availability, but it is not high in maturity. FEAF has good Reference Model Guidance, Partitioning Guidance Prescriptive Guidance, but it is weak in Business Focus and Time value. That is the reasons for FEAF not being used in current business. Gartner Framework has good business focus and Time to Value, but is weak in information availability. Therefore, the top 4 EAFs have different characterises and specialisations.



	Ratings			
Criteria	Zachman	TOGAF	FEA	Gartner
Taxonomy completeness	4	2	2	1
Process completeness	1	4	2	3
Reference-model guidance	1	3	4	1
Practice guidance	1	2	2	4
Maturity model	1	1	3	2
Business focus	1	2	1	4
Governance guidance	1	2	3	3
Partitioning guidance	1	2	4	3
Prescriptive catalog	1	2	4	2
Vendor neutrality	2	4	3	1
Information availability	2	4	2	1
Time to value	1	3	1	4

Figure 2.13 – Criteria and Rating for each EAF (Sessions, 2007)

Therefore, after reviewing the advantages and disadvantages of available frameworks, TOGAF has been selected in this project as the basis from which new methodologies are developed so that knowledge can also be managed by IT systems in the product development process for the following reasons: (i) TOGAF can help enterprises build their own EAF with their own *dynamic* processes; (ii) TOGAF considers both business requirements and information/knowledge users' requirements in a *holistic* way; (iii) TOGAF also provides *development methods* for each stage. However, how to manage enterprise knowledge has not been addressed sufficiently. TOGAF has not been particularly used in the product development process either. Therefore, the aim of this research project is to investigate the potential of the powerful Enterprise Architecture Frameworks for managing the important and intensive product development knowledge.

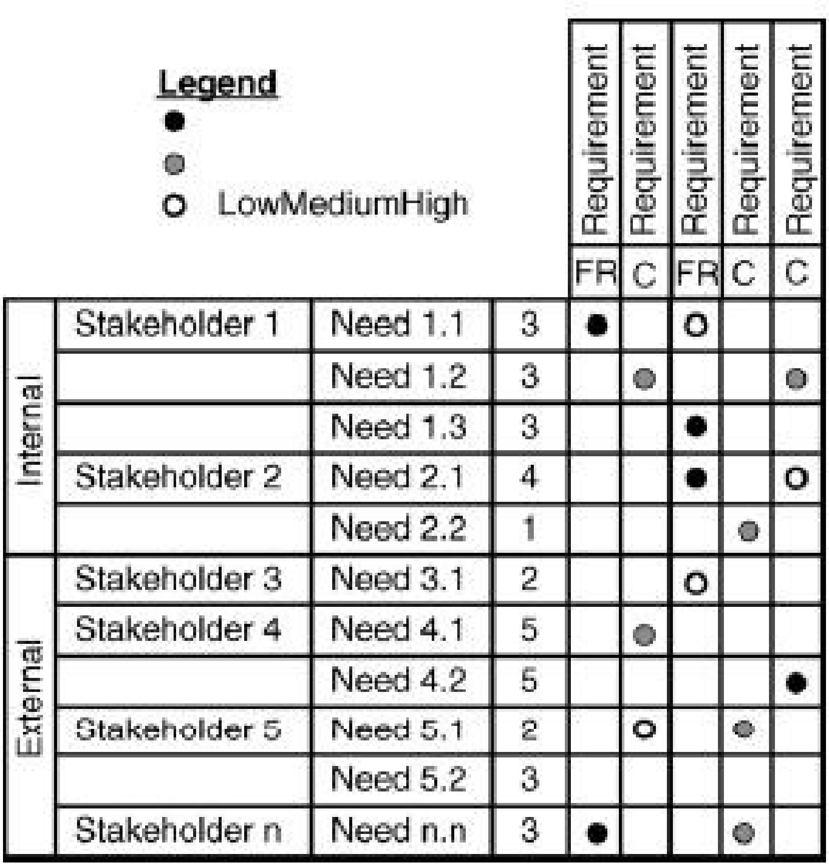
2.6. *Requirement Management*

The methods and tools previously developed in the requirement management area are suitable for both business requirement and knowledge user's requirement in this project. In this section, the concepts and previous research in requirement management will be discussed.

In the current industrial environment, product development efforts are focusing on meeting customer requirements. Many researchers investigated and claimed that requirement management becomes more important in global competition. Chatzoglou (1997) claimed that requirement management can improve the performance of enterprises, particularly in the software development business. Chatzoglou developed a methodology to prove this. Emam and Madhavji (1995) proved that requirement management can help enterprises save cost and time in software development. A good requirement lifecycle leads to a good requirement engineering and management in enterprise. Davis and Zowghi (2006) specified a 6 step requirement lifecycle consisting of gathering, analysing, selecting, documenting, verifying and managing. This 6 step lifecycle provides a method for enterprises to manage requirements.

Baxter et al (2007) developed an integrated product model which contains 5 main activities which represents the procedure from the customer requirements to engineering requirements. According to this model, the requirement process is clearly illustrated. Firstly, the customer requirements are captured in the current market. At the same time, the technical characteristics to meet customer requirements are captured from internal experts. After these, the customer requirements are analysed and transferred to engineering specifications. The engineering specifications form parameterised specifications which are the engineering requirements in the enterprise. The knowledge is then stored in the knowledge reuse model. This model provides knowledge management capabilities to help enterprises manage their requirements as knowledge. However, this model does not consider the knowledge users' requirements in the design process.

Khurana and Rosenthal described the failure at the fuzzy front end stage of product development if company's strategy and all stakeholders' requirements are not considered properly (Khurana and Rosenthal, 1998). Nilsson and Fagerstrom (2006) developed a methodology to manage stakeholder's requirements. Firstly, the relationship of the product being developed with different stakeholders needs to be identified. Then use some methods to determine stakeholders' needs, such as interviews, questionnaires and focus groups (Bender et al, 2002). Then establish a stakeholder and requirement matrix. An example matrix is shown in figure 2.14. Finally, the various requirements need to be balanced, in order to find a best and compromised solution.



Legend
●
◉
○ LowMediumHigh

				Requirement	Requirement	Requirement	Requirement	Requirement
				FR	C	FR	C	C
Internal	Stakeholder 1	Need 1.1	3	●		○		
		Need 1.2	3		◉			◉
		Need 1.3	3			●		
	Stakeholder 2	Need 2.1	4			●		○
		Need 2.2	1				◉	
External	Stakeholder 3	Need 3.1	2			○		
	Stakeholder 4	Need 4.1	5		◉			
		Need 4.2	5					●
	Stakeholder 5	Need 5.1	2		○		◉	
		Need 5.2	3					
	Stakeholder n	Need n.n	3	●			◉	

Figure 2.14 – Stakeholder and Requirement Matrix (Nilsson and Fagerstrom, 2006)

Some requirements do not come from existing customers. They are from potential new customers and from relevant policy makers. Therefore, a comprehensive product development process should be driven by all stakeholders' requirements.

In this project, Knowledge users' requirements and business requirements are the main driver of IT/KM system design. Karlsson et al (1996) claimed that one serious problem in product development is that the product does not satisfy customer requirements and expectations. The 'products' can be the traditional physical products, and can also be software systems. The captured knowledge users' requirements need to be fully analysed and understood before starting to develop information system specification (Ko et al, 2007). Any large engineering projects have various individual requirements to be considered (Firesmith, 2004). It takes a lot of time and resources to translate the individual users' requirements into system requirements.

In practice, not all users' requirements need to be transferred to system specification. Requirements prioritisation methodologies are used to identify the most important requirements (Sommerville, 2006). Many researchers reported the importance of requirement prioritisation. Karlsson et al (1996) described that only 36% functions of IT systems are used, the rest of the functions are rarely or never used in real-life applications. Sommerville (1996) categorised the requirements into three categories: essential requirements which must be included in the system, useful requirements which could be reduced if the system or project is out of the boundaries, and desirable requirements which are optional and should be made available at special request.

There are a number of requirement prioritisation methods being used in practice, and they are based on one of the two types of criteria, i.e., (1) the mathematical formula and comparison method such as Analytical Hierarchy Process (AHP), Case-Based Ranking (CBRank) and Binary Search Tree (BST); and (2) the human voting or human categorising methods such as 100-Points method and Planning Game (PG). The main advantage of AHP is accuracy. AHP is a mathematical formula which can be easily understood and used. It can avoid the largest source of errors, i.e., the human judgement with their limited knowledge and experiences (Saaty, 1980). AHP method compares all requirements each other. In a large project, there are thousands of requirements. This means that there would have thousands of comparisons. Therefore the AHP method can be too complex for use in the initial stages of large projects.

100 Points method is one of the oldest and simplest methods for requirements prioritisation (Leffingwell and Widrig, 2003). It is also called cumulative voting. All relevant people, including project team members, representatives of customers and suppliers, vote on each requirement from 1 to 100. Then, the points of each requirement are summed up. The requirement with the highest points is the most important. The main advantage of this method is that it is easy and quick to get the results. However, the accuracy is not good. The points of a requirement are influenced by the different backgrounds of the voters. It is not objective compared with the mathematical method.

Planning Game is a popular method recently. All requirements are divided into three by customers: 'those without which the system will not function', 'those that are less essential but provide significant business value' and 'those that would be nice to have' (Beck, 2001). At the same time, the designers and programmers divide all requirements into threes based on risks, i.e., 'those that can be estimated precisely', 'those that can be estimated reasonably well' and 'those that cannot be estimated at all'. The requirements in both the firsts are the highest priority. The Planning Game method is very flexible. It can achieve high accuracy, and is relatively quick and easy to prioritise requirements.

In this project, several requirement prioritisation methods have been used for the determination of the KM system specifications. In other words, the most appropriate requirements prioritisation methods are selected and used in different development areas and stages. For example, according to the advantages and limitations of each method, 100 Points method can estimate which requirements should be considered as top priority in the knowledge system design, because it is fast and simple to use, especially for large enterprises and short development time. The collaborating company is a large company with 20,000 employees and multi-million pounds asset. The knowledge system must cover the whole product development process consisting of at least 4 main departments and several groups. The main concern of 100 Points method is the accuracy. For the early stage of product development, the accuracy of 100 Points is accepted by the stakeholders. In later detail design stages in which the accuracy is the top priority, AHP are selected as the method for the requirement prioritisation.

The requirements transferring process is the basis of the whole knowledge system development. An appropriate process can lead to efficient development procedure in IT system development. The adopted requirement transferring process will be fully described and discussed with a real industrial example later.

2.7. System Development

In this section information system development life cycle will be discussed, in order to find an appropriate methodology for this project.

Blackchard and Fabrycky (1990) defined system development life cycle as a process for creating system and the methodology which system developers can use to develop their own systems. They also claimed that system development life cycle can help organisations to solve their complex system development problems. Haag et al (2001) defined system development lifecycle as structured step-by-step approach for developing information systems. They developed a 7-steps system development life cycle including: (1) Planning which defines what system to be developed and project plan including tasks, resources and timescales; (2) Analysis which captures business requirements for the system development; (3) Design which designs system models and techniques to support the system; (4) Development which builds the database, programmes and technical architecture; (5) Testing which identifies testing conditions and performs testing of the system; (6) Implementation which writes detailed user documentation and provides the training of users; (7) Maintenance which allows systems changes and builds a help desk for users. This 7-steps product life cycle is widely accepted in industry.

However, many researchers claimed that different companies should have different system development life cycles (Chapman et al, 1992; Kerzner, 1995). Haag et al (2001) indicated that the 7-steps system development lifecycle is not suitable for every industries based on current industrial environment in their further research. The 7-steps system lifecycle does not consider users' requirements.

Bahill and Dean (2009) claimed that system development must consider the requirements of customers (users) and other stakeholders. In this terms, customers and stakeholders include anyone has right to impose the requirements on the system. Bahill and Gissing (1998) defined a 7-steps system development life cycle to ensure the customer's needs are satisfied throughout this entire system development lifecycle. The 7-steps life cycle includes: (1) specify the problem: identifies customers and understands customers' needs; (2) investigate alternatives based on cost, performance and risk; (3) model the system: models the clarified requirements, possible bottlenecks and duplications; (4) integrate designed interface and system elements and then tests them as a whole system; (5) launch the system: it runs system and produces output, in order to finish system development; (6) assess performance: performance is evaluated with evaluation criteria, in order to measure the system; (7) re-evaluation: it aims to create a continual and iterative process for further development. It is a big advance for system develop life cycle.

However, the 7-stepts lifecycle is not suitable entirely for this project. Although it considers customer's needs, just understanding customer's requirements is not enough for developing knowledge management system. Both enterprise objectives and knowledge users' requirements need to be considered in the system development methodology. Knowledge users' requirements should be analysed and prioritised before using them to develop system specification. Therefore, a formal knowledge management system development process is needed, in order to provide a guideline which system developers can follow.

System development methodology is used to structure, plan and control the process of developing an information/knowledge management system (Knight et al, 2003). The Center for Medicare and Medicaid Services (CMS) of U.S. Department of Health and Human Services (2005) provided a full analysis of current popular 5 system development methodologies including Waterfall, Prototyping, Incremental, Spiral and Rapid Application Development. The analysis includes types of methodologies, principles, strengths, weaknesses and appropriate solutions for each methodology.

CMS claimed that the Waterfall methodology is linear, and it is ideal for less experienced project manager. However, it is inflexible, because of the linear structure. It is hard to respond to changes in the methodology. It needs a set of full understandable requirements to support its application. The Prototyping methodology has iterative structure in system design, coding and testing. Therefore, it provides an opportunity for investigating users' requirements iteratively when the requirements are not clear. It may lead to uncertain cost and time period in the iterations of identifying users' requirements. The Incremental methodology combines linear and iterative structure, in order to reduce inherent project risk by breaking project into segments. It can ensure minimised risks and maximised successful rate of system development. The main weakness is that if some modules must be finished earlier than others, it may lead to delays when these modules can not be finished in time. The Spiral methodology also provides a combined linear and iterative structure. The working principle is similar to the Incremental methodology. But it provides more ease-of- changes in the development process. It enhances risk avoidance. However, it provides a very complex and changing process. It requires an experienced project manager to use. The Rapid Application Development methodology has an iterative structure. It can provide a fast and high quality system development with lower cost. It also breaks project into segments, in order to reduce project risks. However, it is hard to reuse its developed models, because of the complex developing procedure. It may lead to a higher cost in future development when the key personnel left the project.

All these 5 system development methodologies have their strengths and weaknesses. In Knowledge Management system design, the continual development method is one of the necessary aspects. Therefore, the principle of Rapid Application Development is not suitable for this project. The iterative structure provides an excellent opportunity for investigating uncertain users' requirements. Therefore, the linear structure is not suitable for this project. The Waterfall methodology is not suitable either. One characteristics of TOGAF is its iterative structure which makes TOGAF suitable for any size of enterprises. Even departments and groups of an enterprise can be model with TOGAF. Therefore, the Increment methodology and the Spiral methodology may lead to a redundant and complex structure in the proposed methodology. Finally, the structure of the Prototyping methodology is suitable for this project. It provides an iterative structure for requirements definition, system design, coding and testing. This structure (as shown in Figure 2.15) can ensure the identification of comprehensive users' requirements and system specifications through the iterative procedures, which is the necessary aspect of a Knowledge

Management system development methodology. In this project, the initial investigation, requirements definition and system design of the Prototyping methodology are used to develop the proposed methodology and to analyse the requirement involved in. It should be noted that this project does not follow the complete process of the Prototyping methodology, as coding and implementation is outside the scope of this project. However, users of the methodology developed in this project are recommended to follow the Prototyping methodology for system development.

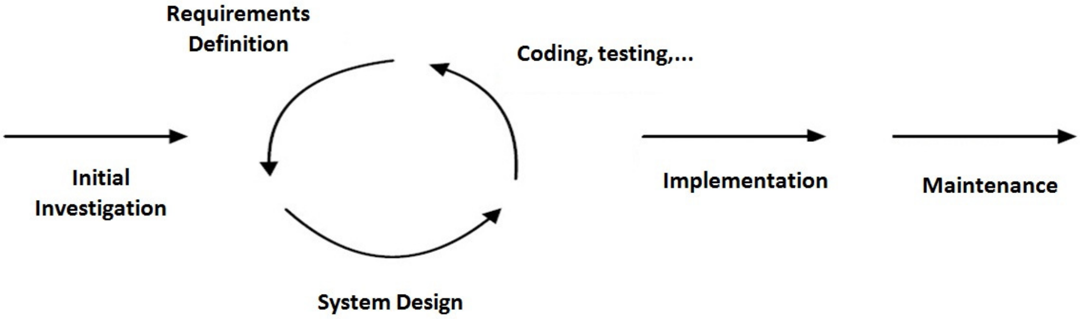


Figure 2.15 – The Structure of Prototyping System Development Methodology (CMS, 2005)

2.8. Summary

In the literature survey, the related work in the proposed area has been investigated in order to identify the gaps of current research and development. The research direction of this project can then be defined based on the gaps. In the new product development domain, there are various IT systems available to manage product development such as PLM systems. However, most IT systems do not have knowledge management capabilities. There are growing efforts devoted to enterprise knowledge management, but there are still no formal methodologies to follow in knowledge management system design. Formal methodologies have been developed by researchers and practitioners for the design and implementation of IT systems for the management of enterprise information, such as Enterprise Architecture Frameworks. However, these frameworks are not designed for managing knowledge. Therefore, a formal methodology, which can manage information as well as knowledge in product development, is required. The main aim of this project is therefore to develop a methodology for knowledge management system design based on Enterprise Architecture Framework. The methodology should be driven by both knowledge users’ requirements and business objectives in the manufacturing business.

Chapter 3: Industrial Investigation

Industrial investigation is very important and becomes a necessary part in every manufacturing related research. Usually, the main aim of industrial investigation is collecting data from companies, and then verifying and evaluating the developed methodology, theory or technologies with real industrial examples. In this research, a proposed knowledge management system design methodology is developed based on the collected data from the collaborating company. This chapter will describe the procedure, methods and results of data collection in the collaborating company, and will introduce the collaborating company and why this company is suitable for this project. The management style of the product development in this automotive company will be described. The organisational structure of the automotive company will also be described. The business process of the product development project will be described which will be the business architecture of the proposed methodology. Two kinds of requirements in the company will be identified, i.e., business requirements and knowledge user's requirements for knowledge management system. Both kinds of requirements will be discussed in this chapter.

3.1. *Introduction to the Collaborating Company*

In this research, the whole process of methodology development is based on real industrial cases. An automotive manufacturing enterprise is the collaborator of this project. It is a new company and relies heavily on its advanced new product development process. The total asset of the company is around £1b pounds, and the total employees are over 20,000. The main products of this company are tractors, trucks, platform trucks, buses and sports utility vehicles (SUV). Some example products are shown in Figure 3.1.

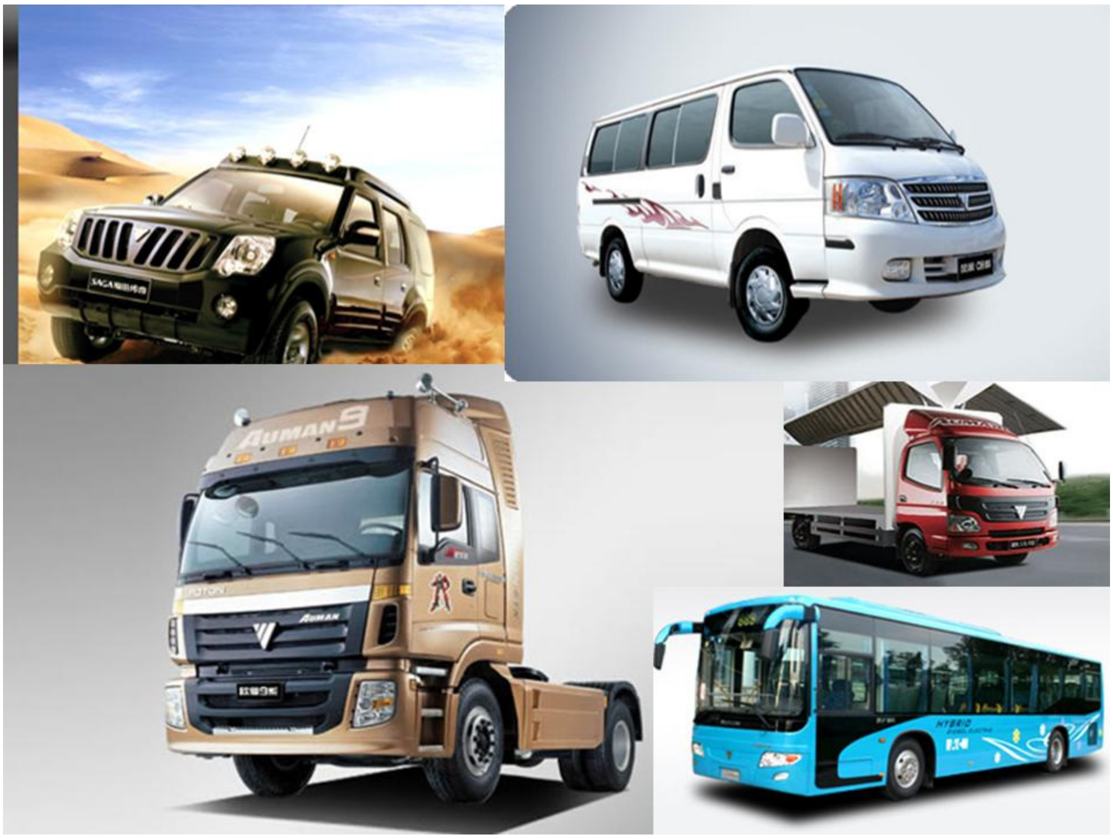


Figure 3.1 – Example Products of the Collaborating Company (contest of the company)

The automotive company focuses on product innovation which is one of its corporate strategies. There are three types of product innovation: original innovation, integrated innovation and innovation based on existing products. Therefore, the development mode of this company is to integrate knowledge and uses it in innovation of new product. Therefore, the research theme of this project matches the strategic requirement of this company very well.

3.2. Objectives of Industrial Investigation and Method Used

The first objective of the industrial investigation is to understand the company's organisation structure. In the context of this project, the organisation structure is the departments and groups that contribute to or are involved in the product development project. The methodology to be proposed in this project regard the product development project as an enterprise, the organisational structure can thus be divided into 3 levels: project level (enterprise level), functional level (department level) and task level (group level). One of the advantages of TOGAF is that each level of the organisation can be modelled based on the ADM iteration, and then the different levels can be integrated so that the complex structure of the enterprise can be systematically managed. The second objective of this investigation is to understand and capture the business process which is the product development process in this project. TOGAF can help enterprise to build the Business Architecture based on the existing business process. The objective is to understand and capture knowledge users' requirements in the new product development process. The knowledge users are managers and engineers involved in product development, and their requirements should be transferred to system specifications through a requirement transferring process including analysis, categorisation and prioritisation.

There are a number of methods that can be used for data collection, such as interviews, questionnaires, observations, documents review and group discussions. In this research project, data need to be collected from the top management to technical engineers of the enterprise. The relevant data are comprehensive and might be anywhere in the enterprise. Therefore, finding the appropriate roles in the enterprise is important to ensure the quality of the results of the data collection. Therefore, face to face interviews including meeting discussions based on semi-structured questionnaires are adopted as the data collection method in this research. A sample questionnaire is shown below and the full version will be provided in Appendix A:

Questionnaire for designers and project mangers

Name:
Date:
Position:

- General Questions
    a. What type of product do you develop?
    b. Could you describe your work scope and responsibility?
    c. How many people are there in your team?
- What are the business objectives of a new product development in your company?
- Is the new product development driven by requirements? If yes, please describe how requirements drive the development process.
- How does your company capture, specify and use the various requirements in new product development? Such as customer requirements, business requirements and engineering requirements.
- Is there any knowledge based engineering system being used in your company? If yes, please describe how to capture, store, share and reuse knowledge in your department.
- Are there any enterprise guidelines or processes as a standard to guarantee the company achieve the business goals and objectives? If yes, please describe it with its architecture.
- How does your company keep product data? Are there any descriptions of each data in database? E.g. drawings, CAD Data, etc.
- Are there any KM or IT systems being used in your company? If yes, can you please introduce the system you use?
- What kind of ideal KM system do you need based on your position and your team?
- Are there any special requirements of your team for the system?

In this data collection, most data is not numeric. Many processes and structures need to be collected and understood. The face to face interviews and meeting discussions help interviewer understand these processes and structures during the interviews and meetings. One advantage of semi-structured questionnaires is the flexibility. Semi-structured questionnaires have a basic structure, but the detailed questions can be asked by the interviewer based on the situations and requirements. It is appropriate for this research.

In this research, many data collections include knowledge from knowledge users such as the managers and engineers from each department. The new product development process consists of several activities such as analysing, planning, designing, manufacturing, sales and marketing. All requirements of these knowledge users need to be fully understood. Therefore, the manager responsible for these activities need to be interviewed. Some activities consist of more than one department, and each department manager need to be interviewed. The department managers form the knowledge capturing at the department level. In the group level, each group leader will be interviewed, in order to capture the relevant knowledge involved in each group. At the top level which is the enterprise level, the collaborating company assigns product managers to manage product development projects. If any product development project is viewed as an ‘enterprise’, the product manager responsible for that project is the best interviewee for capturing the relevant knowledge at enterprise level. The brief description of product manager in the collaborating company will be discussed in the next section.

3.3. The Role of Product Manager in the Collaborating Company

The investigated automotive company was founded in mid-1990s. It is a relatively new company compared with other international automotive enterprises. When it was founded, one of the main strategies is product innovation. Therefore its management structure of new product development is organised in a similar way as existing companies in the same business. Each individual product development project is managed by a role named Product Manager who is a member of the top management (board of directors), e.g., vice-president. The Product Manager has enough power to plan, design and integrate products directly, and directly communicate with chair of the board. Another main responsibility of the Product Manager is to manage new product development as an individual project. The Product Manager controls cost, process, investment, benefits, quality and performance in the product development process.

The role of the Product Manager in the collaborating company is more than the traditional Project Manager in manufacturing enterprises. Usually, a traditional Project Manager just controls the new product development as one single project, and he/she handles simple products well. However, the Project Manager does not have enough power during large and complex projects, particular in big decision making, assessment of people’s performance and personnel management, since his/her project team members are drawn from different departments. When the cooperation of people from different departments has some problems, the Project Manager does not have enough power to handle it without consulting to their departmental heads. This situation may lead to longer development time and more cost. Another possible problem is that the horizontal communications among departments at the same level are weak. The Project Manager is good at vertical communication in the project, but in horizontal communications, the Project Manager is weak. It may influence the main objective of this project. Product knowledge loss is another problem due to the lack of authority of the Project Manager in personnel management. Good knowledge management should be based on a stable and self-giving communication environment which contains a matrix structure with both vertical and horizontal communications.

Therefore, in the large and complex product development projects such as automotive products, the senior role of Product Manager (who is a board member) is a good arrangement compared with the less senior traditional Project Manager in managing large and complex product development projects. Therefore, the Product Manager is the best person to capture knowledge at the enterprise level.

3.4. The Organisation Structure of the Automotive Company

According to the Architecture Development Methodology (ADM) of the Open Group Architecture Framework (TOGAF), the knowledge manage system needs to be built and designed based on an organisation structure of the enterprise with associated business processes. In an automotive company, the whole product development process consists of most departments of the enterprise. The main functions of several departments largely follow the main stages of similar product development processes in the automotive business. Some departments play key roles, such as the Design Department, and some departments play supporting roles, such as the Finance Department.

As product development is selected as the focus for applying the proposed methodology, the whole product development activities can be regarded as a design project, and in TOGAF terms, can be viewed as an individual enterprise. The organisational structure (as shown in Figure 3.2,) represents an affiliation with a parent and children relationship like a family tree with three "generations" (levels). The first level is the enterprise level, i.e., the product development enterprise. The second level is the department level which includes five departments involved in different stages of the product development process.

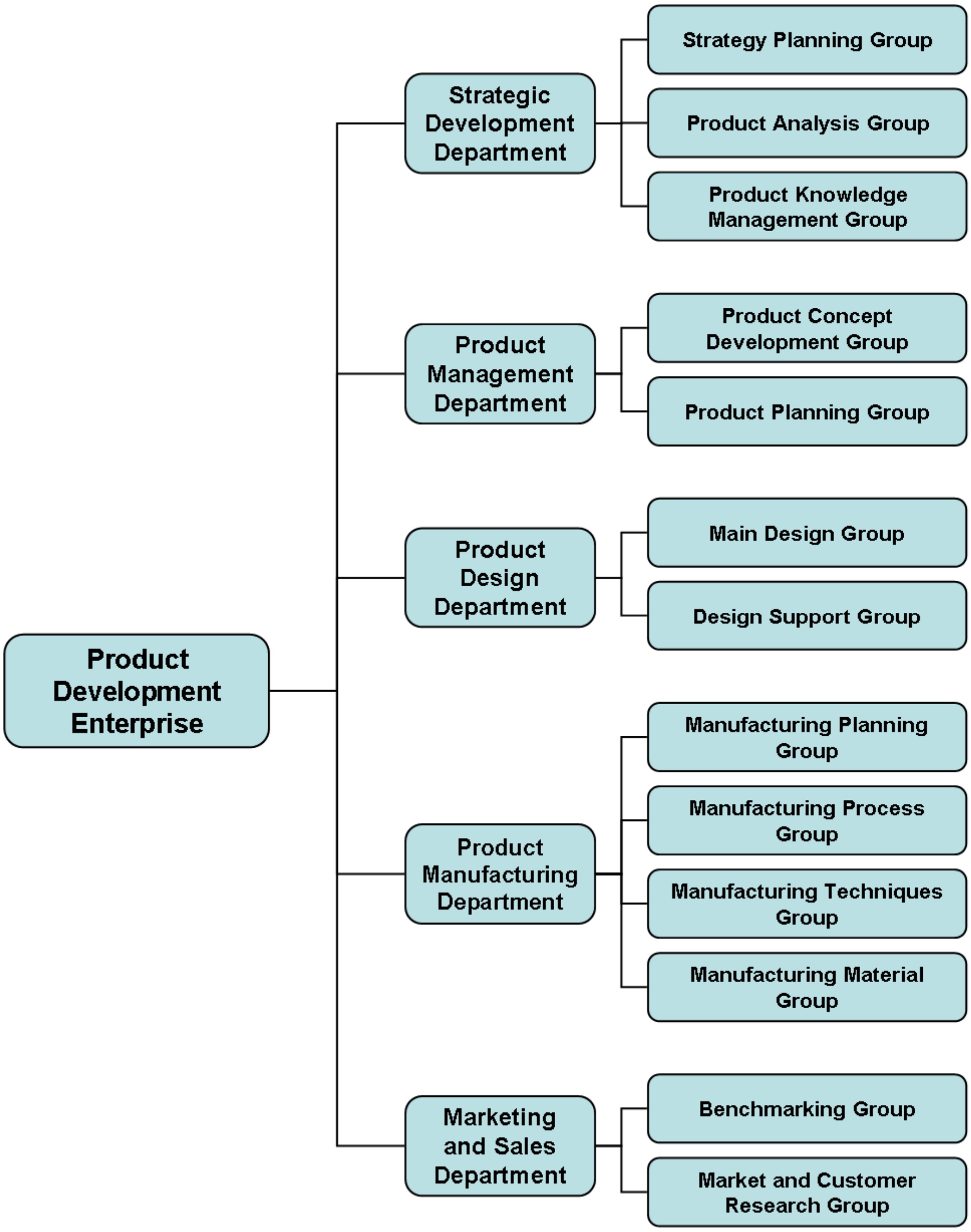

Product Development Enterprise
Strategic Development Department
Strategy Planning Group
Product Analysis Group
Product Knowledge Management Group
Product Management Department
Product Concept Development Group
Product Planning Group
Product Design Department
Main Design Group
Design Support Group
Product Manufacturing Department
Manufacturing Planning Group
Manufacturing Process Group
Manufacturing Techniques Group
Manufacturing Material Group
Marketing and Sales Department
Benchmarking Group
Market and Customer Research Group

Figure 3.2 – Organisation Structure of the Automotive Company

The first department is the Strategic Development Department which mainly analyses the product perspective and gather the needed knowledge for product development. It consists of three groups: the Strategy Planning Group, the Product Analysis Group and the Product Knowledge Management Group. The second department is the Product Management Department which carries out product planning. It has two groups: the Product Concept Development Group and the Product Planning Group. The third department is the Product Design Department which carries out the main design tasks. It has two groups: the Main Design Group and the Design Support Group. The Main Design Group is responsible for car body framework design and engine design. The Design Support Group communicates with others departments and groups to find the correct resource to support the Main Design Group. The fourth department is the Product Manufacturing Department which manages the whole manufacturing processes. This department carries out product manufacturing planning, testing, refinement and production ramp-up. It consists of four groups: the Manufacturing Planning Group, the Manufacturing Process Group, the Manufacturing Techniques Group and the Manufacturing Material Group. The fifth department is the Marketing and Sales Department. Its main function is to gather feedback from existing customers, investigate the potential market information, and benchmark the product with main competitors. It has two groups: the Benchmarking Group, and the Market and Customer Research Group. Based on the proposed methodology, the organisational structure is used as a basis for dividing knowledge into different levels in the ADM of TOGAF. This will be illustrated in the next few chapters.

3.5. Business Process in the Collaborating Company

TOGAF can help enterprises build up 4 architectures – business architecture, data architecture, application architecture and technology architecture. The business architecture is the basis of these 4 architectures. The other architectures need to be built up based on the business architecture. The business architecture also considers the business objectives for the KM system. Therefore the business architecture is the most important architecture in TOGAF, i.e., the business architecture is the foundation of the knowledge management system of a company based on TOGAF. The business architecture is constructed based on the actual business structure or business process of enterprises.

As shown in Figure 3.3, the product development process of the automotive company has a Product Analysis Activity. This Activity is the start and the centre of the process, which is managed by the Product Manager. It links to the Marketing and Sales Activities which collect information from competitors, market and customers. The Market and Sales Activity captures the knowledge such as product characteristics and market share of competitors with benchmarking or other techniques and input the information to the Product Analysis Activity. The knowledge of previous relevant products is integrated in the Marketing and Sales Activity, and the integrated knowledge inputs to the Product Analysis Activity as well.

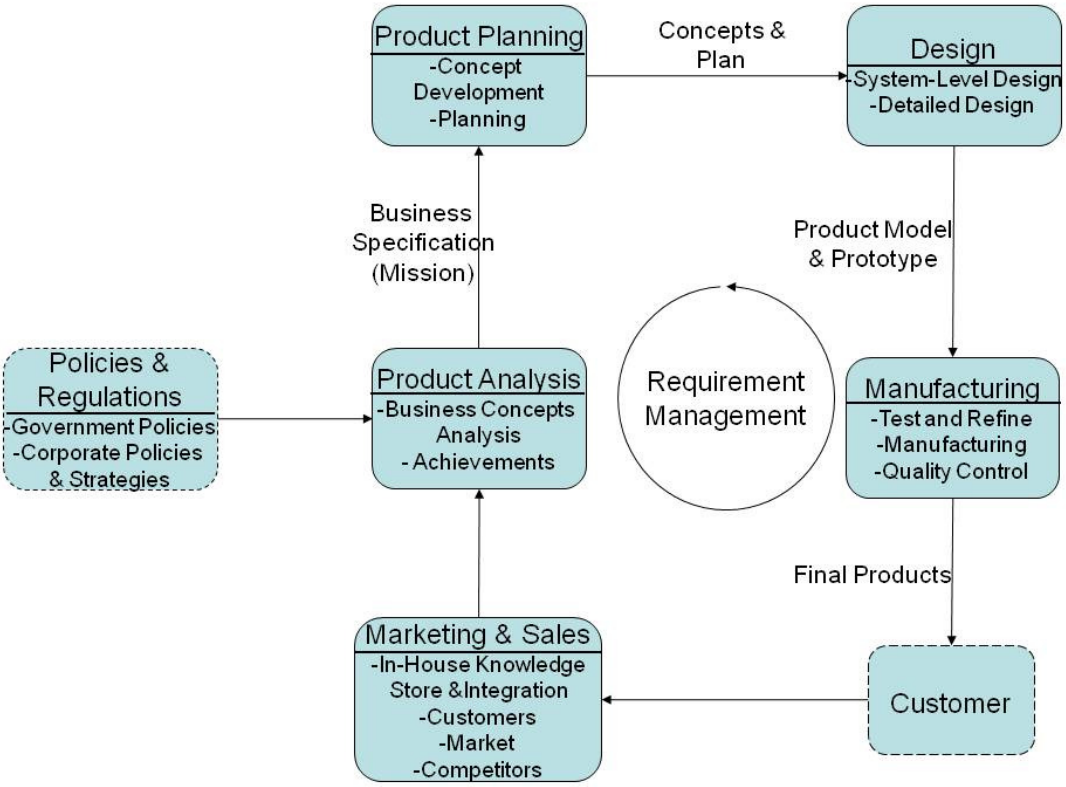


Figure 3.3 – New Product Development Process of the Automotive Enterprise

The government and corporate policies have impact on the current product as shown in Figure 3.3. For example, a new air pollution standard can lead to new engine development. Some strategies in the top management also influence new product development. This product development process is represented as a cycle starting with the Product Analysis Stage, in order to manage knowledge flow in the product development process. The output of the Product Analysis activity is the Business Specification (or mission) which is the main achievement of the product. For example, this new product will bring 3% market share for the company, or this new product will make 30 million pound profits for the company and so on. There are more details provided in the next section.

The next Activity linked to the Product Analysis Activity is the Product Planning Activity. The resulted business specifications (or missions) are the input of the Product Planning Activity. The main objective of the Product Planning Activity is to define several concepts of the new product, including product development process, budget, delivery, warehousing, and production systems. The output of this Activity is the concepts and plans of the product, such as the developed concepts reports, the project plan for the development of product.

After the Product Planning Activity, the Design Activity will start which includes system-level design and detail design. There are some overlaps between each Activity, in order to share knowledge and avoid misunderstanding between activities. The output of the Design Activity is a prototype of the product, and the prototype can be tested and refined in the Manufacturing Activity. After the product is tested and satisfactory, the product is manufactured and delivered to customers through the Marketing and Sales Activity. After this all data and knowledge will be stored and integrated with other relevant knowledge, in order to reuse it in the next generation products.

The product development process is actually a product requirements conversion and classification process through the activities. These detailed requirements determine the functions of each departments and groups. Therefore, in the next section, the product requirements will be discussed.

3.6. Sub-processes of the Product Development Enterprise

According to the new product development process which is shown in Figure 3.3, there is a circle in the development process. This circle shows the requirement management during the process.

The requirement management is the centre of most manufacturing enterprises, particularly in the automotive company. The new product development is based on various requirements, including customer requirement, business objective/requirement, and policy requirement. The majority of new product development is led by requirements pull rather than technology push in the current marketplace. In the collaborating company, when a new product is developed, this product will be viewed as an individual project. The structure of the project team should consist of product manager which manages the whole product development, the leaders of each department which is included in the product development process and the groups of each department.

There are 5 main activities included in the cycle of the new product development process in the collaborating company (as shown in Figure 3.3). These 5 activities form the main product development process flow. The Marketing and Sales Activity collects customer requirements, information about the potential market and product information of competitors. The collected information is integrated as the market requirements, and then it is represented by Product Manager to the board of the company. The Product Manager determines the business objectives and feasibility of the product with the board. For example, the company requires to obtain 3% more market share with this product, the market segment of the product is small family, the target cost is 5,800 pound, the number of units is 100,000 per year, the target price is 8,000 to 10,000, the potential profit is 2,200 to 4,200 per unit, the product will bring 300,000 to 500,000 customers to the company, the product expected to reduce 1-2% market share of the main competitors, and so on. The business objectives are specified and then become business specifications which can be defined as a specified list contained all business requirements. The business specification will be inputted to the next activity. The Product Planning Activity develops a detailed development plan for the product as an individual project. The engineering requirements are developed based on the business specification and customer requirements, and then the engineering requirements can be used to develop the product concepts. The engineering requirement is a basic requirement for design of the product. For example, the inner decorated component should be luxurious for the passengers. It influences the concepts of the product directly. The development plan and product concepts form the engineering specifications which describe details of the product, and then deliver to the Design Activity. In the Design Activity, the system-level design and detailed design are carried out, and then the product model and prototype can be generated. The product model needs to be tested and evaluated, and then it can be inputted to the Manufacturing Activity to produce the product, and then the product can be delivered to customers. The product model can be viewed as the manufacturing requirement which is the criteria for the Manufacturing Activity.

In the collaborating company, each Activity can be broken down into sub-processes. The sub-processes are represented by Design Roadmap modelling technique. As shown in Figure 3.4, the Road Map process model represents the sub-processes of the Marketing and Sales Activities. As discussed in the literature review, DR uses rectangles to represent tasks and uses the rounded rectangles to represent the features which contain information and knowledge as the inputs or outputs of tasks. The arrowed lines are used to represent the process flow and link tasks to features. The main aim of the Marketing and Sales Activity is to gather all kinds of requirements and integrate them together in order to support the Product Manager to make decisions on relevant business objectives.

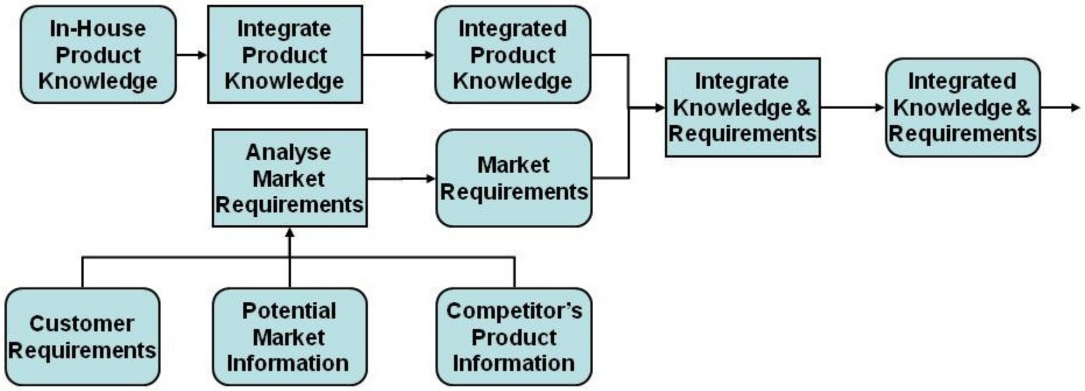

In-House Product Knowledge
Integrate Product Knowledge
Integrated Product Knowledge
Integrate Knowledge & Requirements
Integrated Knowledge & Requirements
Analyse Market Requirements
Market Requirements
Customer Requirements
Potential Market Information
Competitor's Product Information

Figure 3.4 – Sub-Process of the Marketing and Sales Activity

There are two branches in the process. One is the relevant In-House Product Knowledge from the previous product development project. This kind of knowledge is integrated to avoid duplicates and overlaps before combining with the other branches, i.e., the Market Requirements including Customer Requirements, Potential Market Information and Competitors' Product Information. The Market Requirements are combined with the In-House Product Knowledge, and then the final output of the Activity is the Integrated Knowledge and Requirements which are inputs to the Product Analysis Activity.

In the next Activity – Product Analysis Activity, the input knowledge and requirements are converted to possible product ideas through the requirements analysis as shown in Figure 3.5. The product idea is a general image for the product based on market requirements. The product idea usually contains some example products which can be from in-house products or products from the existing market. All possible product ideas are included in the product development, and the most appropriate product idea is selected based on various criteria and constraints.

The constraints are not just some basic factors in the company, such as cost and time, but also these constraints from the government's policies and cooperation policies. For example, a new air pollution standard leads to the new revolution of the engine development. The product idea is evaluated based on these constraints, and then the business specification, missions and objectives are developed based on the product idea, such as aim, objectives and strategy of the product development to be achieved.

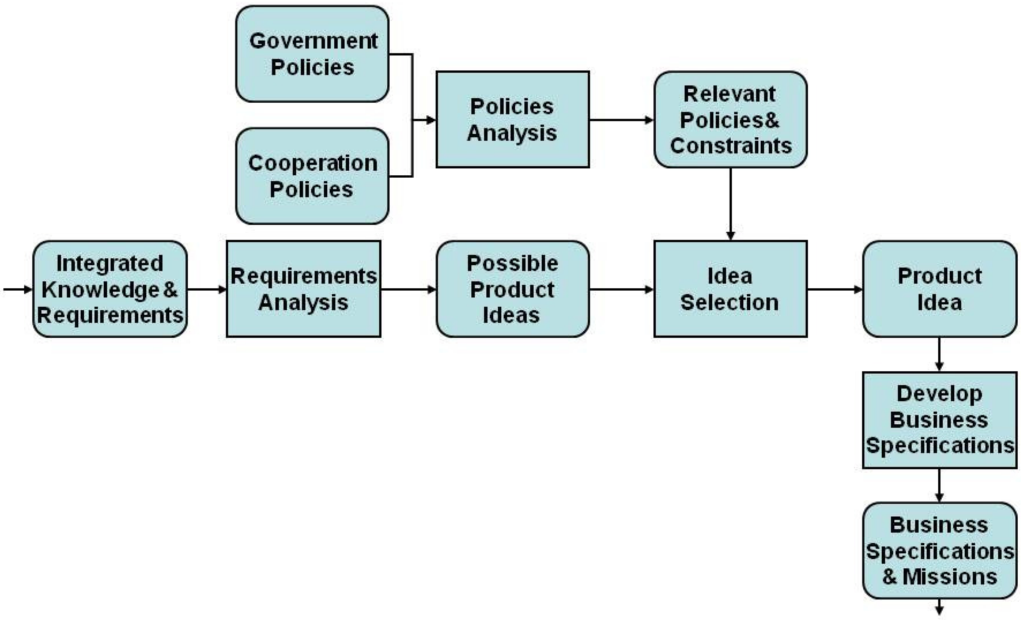

Government Policies
Cooperation Policies
Policies Analysis
Relevant Policies& Constraints
Integrated Knowledge & Requirements
Requirements Analysis
Possible Product Ideas
Idea Selection
Product Idea
Develop Business Specifications
Business Specifications & Missions

Figure 3.5 – Sub-process of the Product Analysis Activity

The next activity is the Planning Activity (shown in Figure 3.6). When the Business Specifications come into this activity, the whole project plan will be developed based on the business specifications and missions. After the project plan is developed, the engineering specification is developed. The engineering specification leads to the product concepts generation. All possible product concepts are developed, and then these product concepts are selected based on some criteria.

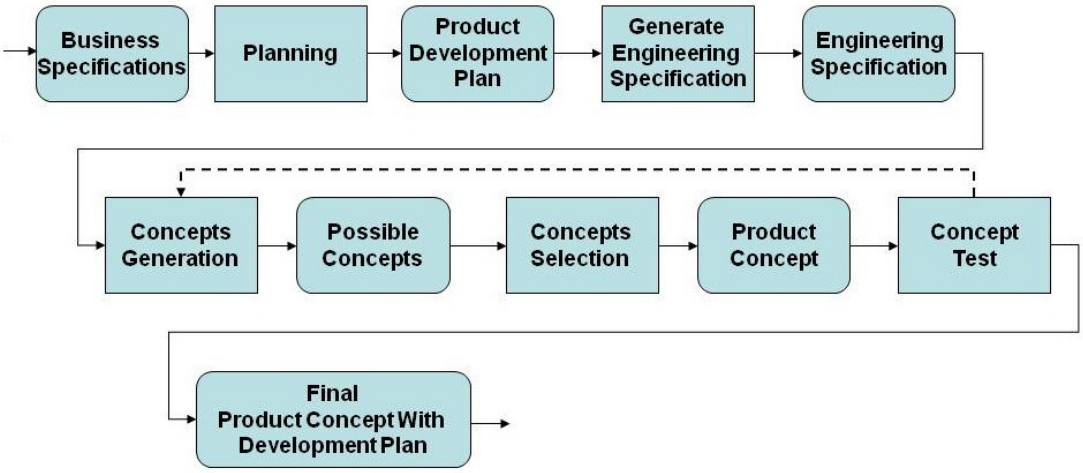


Figure 3.6 – Sub-process of the Product Planning Activity

The appropriate Product Concept is tested, and there is a feedback loop from the Concept Test to Concepts Generation as shown in Figure 3.6. Because if the tested product concept failed, new product concepts are generated and selected again until the product concept passes the test. Finally, the selected concept and development plan are transferred to the next Activity.

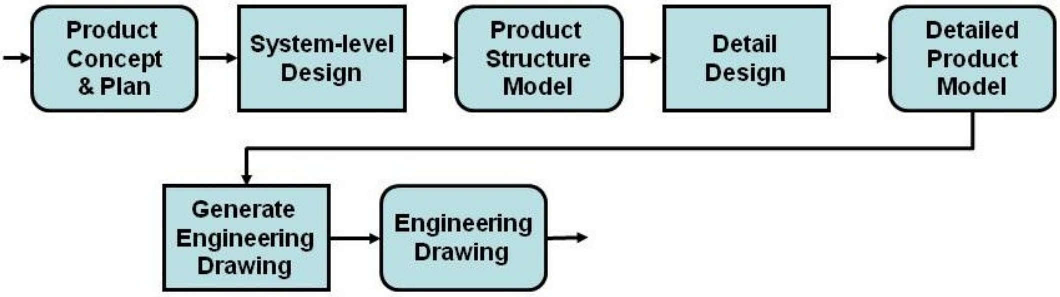


Figure 3.7 – Sub-process of the Design Activity

In the Design Activity (shown in Figure 3.7), the system-level and detailed level design is generated based on the concept and plan. Finally, engineering drawings will be developed in the Activity.

Before manufacturing the product in mass production, prototypes of the components and the assembled product need to produced and tested. As shown in Figure 3.8, when the engineering drawings are inputted into the Activity, the prototypes of components are produced, and then the prototypes are tested. If the prototypes have some problems, they will be corrected until they pass the test.

After the component prototypes pass the test, the components will be assembled to a product prototype for testing. There are two feedback loops in the process of this Activity to deal with the corrections/improvements of the product. After the assembled product passes the test, the product can be mass produced. The In-house Product Knowledge is stored and integrated after project review.

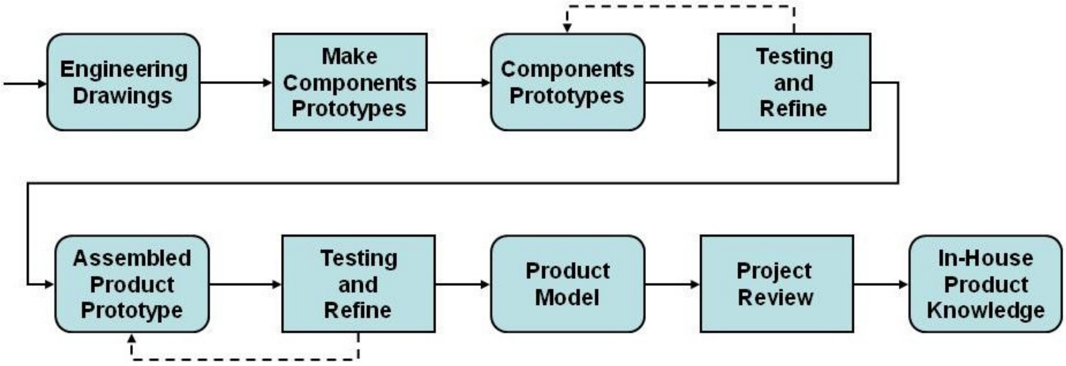


Figure 3.8 – Sub-process of the Manufacturing Activity

In the product development process, the requirements are converted from customer requirements to manufacturing requirements. Each Activity checks the requirements from the previous activity, in order to guarantee less misunderstanding and avoid unnecessary cost and extra developing time. In the next section, the knowledge users' requirements for the knowledge management system will be described and discussed.

3.7. Knowledge User's Requirements for IT Support

In this section, knowledge users' requirements will be discussed based on each department involved in product development. More details will be given in Chapter 7. In the industrial investigation, the results of knowledge users' requirements are based on each department in the product development process. In the TOGAF methodology, the centre of its ADM is requirement management and it links to other stages in the development process. In other words, each stage of ADM needs to satisfy all the these system requirements.

As discussed previously, two main functions of the Product Management Department are product concept design and market analysis. Therefore, one of the main user requirements of KM system is to model the development process, also the business process of departments and groups. Another requirement of the product management department is to collaborate with other relevant departments easily and flexibly. This requirement can lead to more efficiency to the Department. Each Department and Group has its own knowledge user requirements. These requirements are transferred to KM system design requirements, and then system design requirements are transferred to functional requirements of KM system. The KM system can be built based on the requirements of each Department and Group and business requirements.

In the Design Department, the most concerned function of the knowledge management system is the collaborative development and planning with other departments. In the current situation of the automotive company, the Design Department is the main department in the development of the product. Most of the time, the results from the Design Department are evaluated and justified by other departments. If other departments agree with the results, the design will move on to the next step. If not, the Design Department has to make changes and reproduce a new result. It may lead to unnecessary cost and time. The collaborative development and planning can improve this problem.

The Design Department also requires the multiple department cooperation and cooperation. For example, the product concept design needs the Plan Department, Product Profit Department, Purchasing Department, Logistics Department and

Finance Department to work together. It is hard to gather members of staff from each department together to have project meetings.

There are several other requirements of the Design Department, such as knowledge sharing and updating information quickly, designing without using paper, the concept design framework sharing and so on. All of these users' requirements are needed to be transferred to the system design requirements.

The main purpose of the Marketing and Sales Department is to predict the market trend and gathering enough information to support product development. There are 3 domains that Marketing and Sales Department focuses on: customers, competitors and market. Therefore, the requirements of the Department are gathering and integrating information from these 3 domains.

In the customer domain, the Department needs to gather the information from both existing and potential customers, such as new product requirements, product evaluation, new product design based on customers in different regions, the influence of product of competitors and the influence of new relevant policies. All of this information needs to be integrated and stored in the knowledge management system.

In competitor investigation, the Department needs to know the existing product of the competitors, product network, supply chain, value chain, cost, market share, new product plan and characteristics. The information is stored in the knowledge system and integrated with the information from customer and market after the market information is received.

In the market domain, the Department needs to consider the existing market and potential market. When the information is gathered and stored in the system, it will be integrated with other information. Therefore, the main requirement of the Marketing and Sales department is the knowledge gathering and storing. After the knowledge integration, the feasibility report is produced and delivered to the Product Manager. The integrated knowledge is shared in the system with each department in the product development process.

The Product Manager is mainly responsible for product analysis. The Product Manager accepts a lot of information from other departments. The first requirement of the Product Manager is how quickly and precisely to find and invoke the correct knowledge in the knowledge management system. Therefore, the Product Manager needs a standard of the knowledge in the knowledge system based on the first requirement. The Product Manager has to control all the resource in the system. The good visibility of the system is set up for the Product Manager.

To summarise all the different requirements, there are some common requirements of the knowledge system, such as standardisation of the knowledge, multiple cooperation between departments and collaborative development. These system requirements are identified as the high priority of the system design. Some requirements are important in the new product development process. These requirements are also high priority of the system design. The detailed description of the knowledge users' requirements will be given in Chapter 6.

3.8. Summary

The industrial investigation supports this research based on the real industrial data. An important criterion of a development methodology of software and systems is that the methodology can be actually used in real manufacturing enterprises. Therefore, in this research, the whole developing process is based on real industrial cases. An automotive manufacturing enterprise was used as the example. The automotive company is a relatively new company and relies heavily on its advanced new product development process. Before using the proposed methodology to design a KM system, 3 basic elements need to be fully understood. The first element is the enterprise's organisation structure. In the context of this project, the organisation structure is the departments and groups that contribute to or are involved in new product development. The second element is the business processes of new product development projects. The third element is the knowledge users' requirements of each role in the new product development process.

Chapter 4: The Proposed Methodology

This chapter describes the underlying principles of the proposed requirement driven knowledge management system design methodology to support automotive product development. The methodology consists of (i) a guideline (in the form of a flowchart), (ii) an improved Enterprise Architecture Framework based on TOGAF, and (iii) a “Folder-based EAF Implementation”.

In this chapter, the improved Enterprise Architecture Framework will be discussed in detail, including, the reason for using TOGAF, the main concepts of two versions of TOGAF and the links between the constituent elements of TOGAF. The Folder-based EAF Implementation and the guideline will be fully described and discussed in the next chapter.

4.1. The Enterprise Architecture Framework Used in the Proposed Methodology

According to the literature review and industrial investigation described in previous chapters, there is a strong requirement for better understand and manage knowledge in current industrial practice. There is a lack of formal methodologies for designing and developing information system/knowledge management system considering both business and user's requirements. Therefore, this project intends to develop a formal methodology to guide companies to develop information system/ knowledge management system driven by requirements.

First of all, the methodology should be appropriate for both information system development and knowledge management system development. The pure information system without considering enterprise knowledge is no longer satisfying current industrial requirements. Most companies have already used some kinds of information systems to manage their daily work. There is some information/knowledge existing in the current systems. The proposed methodology can be used to ensure what information/knowledge has already been managed by the current system, in order to better manage information/knowledge systematically in further system development.

The process of developing the proposed methodology is shown in Figure 4.1. Firstly, there are some requirements from companies. These requirements are sometime general and implicit. The requirements are investigated through literature survey and real industrial investigations, in order to identify the gaps between industrial requirements and existing technologies. After that, the gaps need to be analysed, and then the proposed methodology is developed. The supporting data of developing the proposed methodology is captured from real industrial environment, in order to ensure that the proposed methodology is suitable for the real industrial applications.

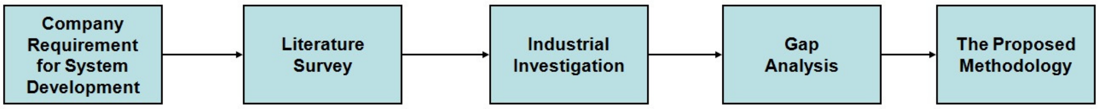


Figure 4.1 – the Process of Developing the Proposed Methodology

According to the literature review, Information Technology (IT) systems have been more widely and successfully implemented in business, compared with the less mature knowledge management systems (Burrows et al, 2005). In real industrial circumstances, a desired system is similar to a workshop which can manage communication, working relationship, people behaviours, knowledge and context of enterprises (Ruggles, 1998, Bush and Tiwana, 2005). After reviewing the advantages and disadvantages of available Enterprise Architecture Frameworks, TOGAF has been selected in this project as the basis from which new methodologies are developed so that knowledge can also be managed by IT systems in the product development process for the following reasons: (i) TOGAF can help enterprises build their enterprise architecture frameworks with their own *dynamic* processes; (ii) TOGAF considers both business requirements and information users' requirements in a *holistic* way; and (iii) TOGAF provides *development methods* for each stage. However, how to manage enterprise knowledge has not been addressed sufficiently in TOGAF, and it has not been particularly used in the product development process either. Therefore, the aim of this research project is to investigate the potential of the powerful Enterprise Architecture Frameworks for managing the important and intensive product development knowledge.

During this project, there are two versions of TOGAF published. As shown in Figure 4.2, according to version 8.1.1, TOGAF provides three main elements, i.e., an Architecture Development Methodology (ADM), an Enterprise Continuum and a Resource Base (the Open Group, 2007). These three elements interact and support each other. ADM provides an architectural process to instruct enterprises to build their information system architectures based on their special requirements. The Enterprise Continuum reflects different levels of abstraction in a development process, and provides a context with continuing construction of the enterprise information system architecture. It will contain two main components: the TOGAF Foundation Architecture which comprises generic services and functions to provide a foundation of enterprise, and the Integrated Information Infrastructure Reference Model (III Reference Model) which is based on the TOGAF Foundation Architecture to help organisations realise information flow. The Resource Base provides technical support on data management, in order to apply ADM. ADM establishes requirements as the centre of an architecture framework. These requirements are information system requirements of each individual groups or activities.

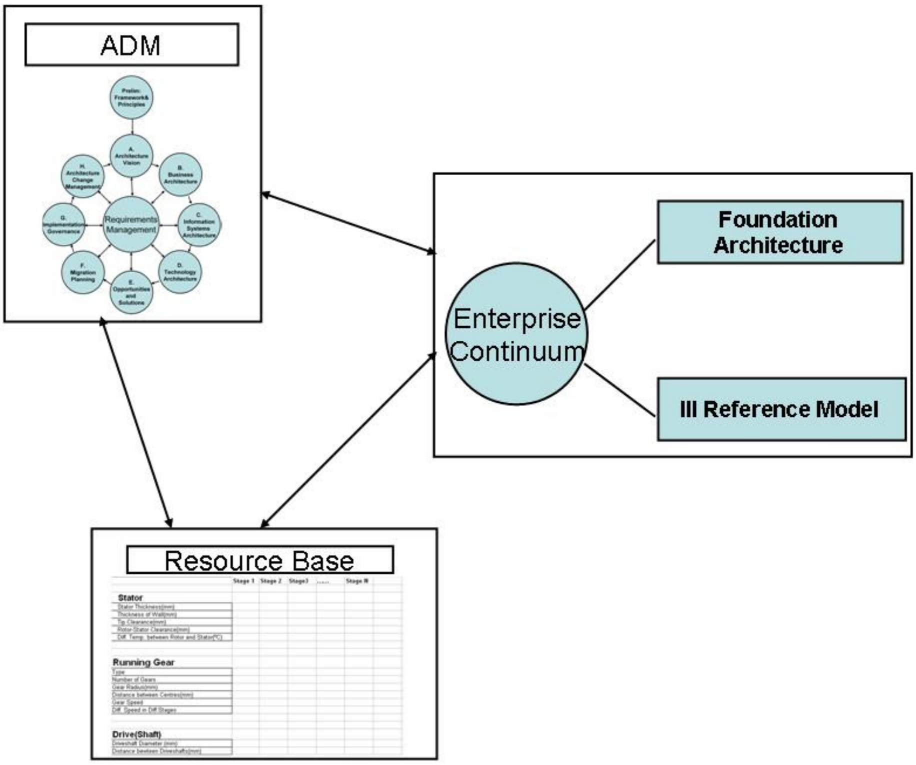


Figure 4.2 – Main Elements of TOGAF (Version 8.1.1)

For example, there is a requirement to develop an enterprise information system architecture in the collaborating company, in order to manage its product information, such as the car-body framework. ADM is the guideline which the automobile company can follow. ADM can guide the automobile company to build its information system architecture based on its business requirements, and then ADM provides the possible solutions and implementation plans. The Enterprise Continuum provides rules and contexts for the enterprise information architecture. The Resource Base provides an information warehouse for managing information, such as the parameters of the car-body framework. All the three elements are necessary in enterprise information architecture development.

In 2009, the Open Group published the latest version of TOGAF which is version 9. In this version, TOGAF enhanced the explanation of ADM which is called

ADM Guidelines and Techniques, in order to guide enterprises to instruct their own EAF easily (the Open Group, 2009). Version 9 (as shown in Figure 4.3) also provides a new element named Architecture Content Framework for better communication between TOGAF and other EAFs. The Resource Base in Version 8 has been replaced with the Architecture Capability Framework. The main features in the previous Resource Base are kept in the Architecture Capability Framework to provide a set of reference material. Other features of the Resource Base of Version 8 are embedded into other elements in Version 9. Business Capability is a representation of ability of the current business in companies. It provides a boundary of the maximum capability of the product development company.

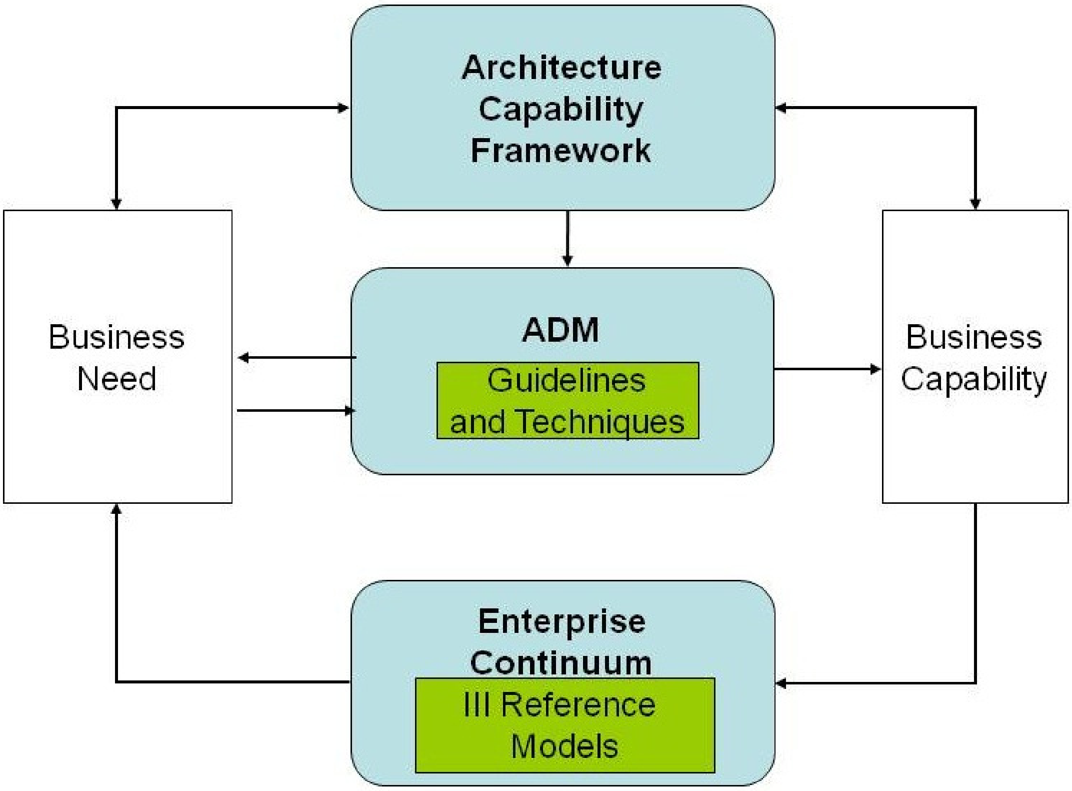


Figure 4.3 – Version 9 of TOGAF

Version 9 of TOGAF is adopted as the basis of the proposed methodology. The main aim of this project is to extend the capability of EAFs so that knowledge is also considered in enterprise information architecture design. It is recognised through investigation that TOGAF Version 9 has the potential to meet the above aim. Knowledge management usually consists of several features, such as knowledge creation, knowledge capturing, knowledge store, knowledge maintenance, knowledge integration, knowledge share, knowledge use and reuse. All these KM features may be integrated with different elements of TOGAF. For example, ADM provides a method and a process to instruct enterprises to create their own knowledge management system with the provided standards and guidelines. Captured knowledge will be managed in the Architecture Capability Framework. One advantage of TOGAF is that it supports continuous system development with its methodology, and this is what KM practice requires. The Integrated Information Infrastructure Reference Model (III-Reference Model), which is included in the Enterprise Continuum, may provide a method for knowledge integration and sharing to support the continuous enterprise system development.

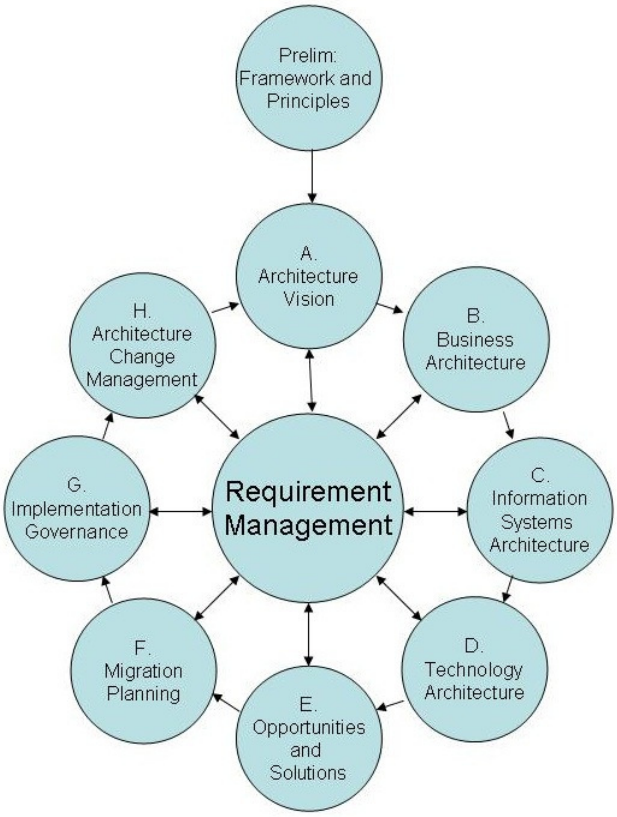

Prelim: Framework and Principles
A. Architecture Vision
B. Business Architecture
C. Information Systems Architecture
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirement Management

Figure 4.4 – ADM of TOGAF (the Open Group, 2009)

However, TOGAF version 9 still has its limitations which should be improved as a methodology for knowledge management system development. In the proposed methodology, requirements are drivers in the ADM (as shows in Figure 4.4). In this project, both business requirement and knowledge user's requirement for knowledge system development are concerned. And this project will focus on product development enterprises. Therefore, the business requirement in this context mainly related to product development requirement. For example, the business requirement is that the weight of a new car-body framework is required to decrease by 5%. The knowledge user's requirement is to have a knowledge management system to make product development more efficiently. Both requirements are managed in the central iteration of ADM (as shown in Figure 4.4).

The Architecture Content Framework is not considered in this project, because its main aim is to provide an opportunity for communicating with other Enterprise Architecture Framework, such as Zackman and DoDAF which are out of the scope of this project.

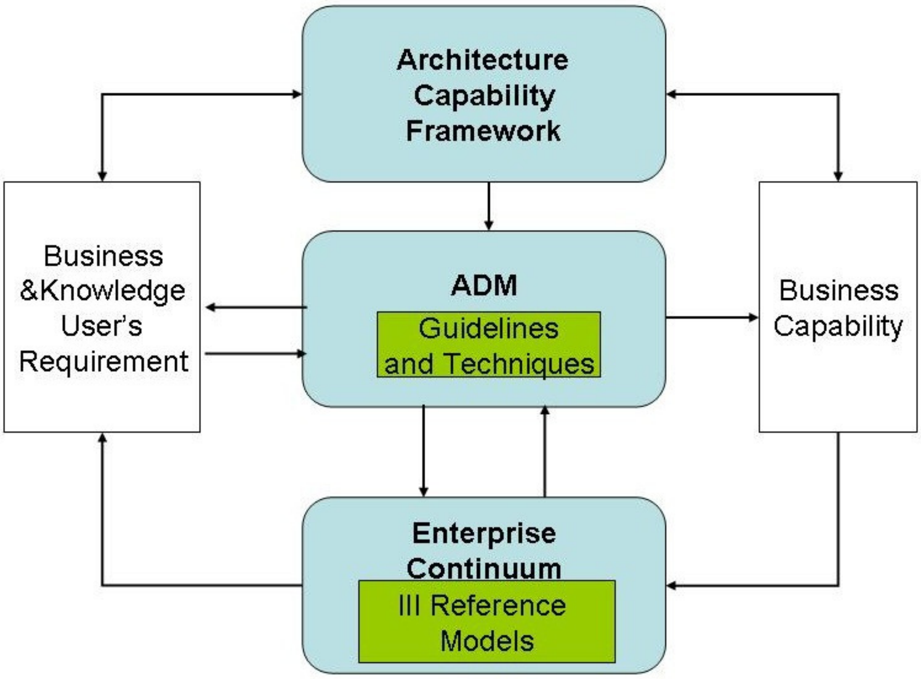

Architecture Capability Framework
Business &Knowledge User's Requirement
ADM
Guidelines and Techniques
Business Capability
Enterprise Continuum
III Reference Models

Figure 4.5 – The Proposed Improved EAF based on TOGAF

Figure 4.5 shows the proposed improved EAF as a main part of the proposed methodology based on TOGAF version 9. One of the main advantages of TOGAF is that each department or group of a product development enterprise can be modelled in ADM iteration or a part of ADM iteration, in order to classify and identify different knowledge user's requirements of each role which is included in the Business Architecture of ADM. The Application and Data architectures which are contained in the Information System Architecture of ADM (see Figure 4.4) can be built based on the specifications which are transferred by knowledge user's requirement. The Technology Architecture of ADM manages the infrastructure and necessary techniques used in the process of building up the KM system. The Opportunities and Solutions iteration manages the possible solutions and results for the TOGAF.

In TOGAF, the Enterprise Continuum will manage virtual assets of the enterprises, such as the knowledge about the context of product generations. It will contain two main elements: the TOGAF Foundation Architecture which comprises generic services and functions to provide a foundation of enterprise, and the Integrated Information Infrastructure Reference Model (III Reference Model) which is based on the TOGAF Foundation Architecture to help organisations realise information flow. The improved EAF adopted similar method for this. For example, the product development knowledge will be managed in the Enterprise Continuum, such as models, processes, design, manufacturing knowledge, marketing data, customer and competitor information.

The Architecture Capability Framework provides a knowledge warehouse of the product development process based on knowledge flow identified in the Enterprise Continuum. The Enterprise Continuum manages the integration and share of knowledge between the three main components. It can be linked with the Application Architecture and Data Architecture of ADM. Knowledge will be stored in the Architecture Capability Framework as knowledge warehouse through the Data Architecture. Architecture Capability Framework also helps the product development enterprise define business targets.

TOGAF version 9 contains Business Need and Business Capability to represent the whole iteration of Figure 4.3. In the proposed methodology, both Business Requirement and Knowledge User's Requirement will be considered (as shown in left of Figure 4.5). More details about users' requirements capturing and analysis are provided in Chapter 6.

ADM can help the enterprise understand the Business Requirements. Business Capability provides a boundary of the maximum capability of the product development company in this single iteration. The solution cannot be outside the boundary of the capability of the company. At the same time, Business Capability can be increased by ADM after finishing this single iteration. Enterprise Continuum can be updated by Business Capability after the relevant ADM process is finished. Enterprise Continuum also can inform the current business status and identify the current Business Requirements based on repository. As the knowledge base, enterprise can define and set targets based on the knowledge stored in the Architecture Capability Framework. Enterprise can identify Business Requirements based on the knowledge already existing in the Architecture Capability Framework. Also, the new integrated knowledge can be stored in the Architecture Capability Framework after the whole procedure. Therefore, the whole structure includes three necessary elements and several iterations to interact among these three elements.

4.2. The Principles of the Architecture Development Method

As explained previously, each department or group of an enterprise can be modelled in the ADM including the enterprise itself. Therefore, an enterprise can be vertically divided into several levels. Using the collaborating company as an example, Figure 4.6 shows three levels of the enterprise. In the top level which is the Enterprise Level, the Business Architecture of ADM consists of the first level of abstraction of the New Product Development (NPD) process (refer to Figure 3.3 for details that are readable in the NPD process). Each Activity at this level consists of a sub-process which is the next level of abstraction. As shown in Figure 4.6, the Design Process in the NPD process consists of 7 design steps, i.e., Planning, Concept Development, System Level Design, Detail Design, Testing, Design Modification and Product Evaluation (refer to Figure 3.7 for details that are readable in the Design Process).

In the same way, each Step in the middle level consists of further sub-steps (activities) corresponding to Group or functional level. ADM method is applied to each level, and the corresponding level of abstraction of information stored in each 'cycle' of the ADM diagram (the left hand-side of Figure 4.6) will be used.

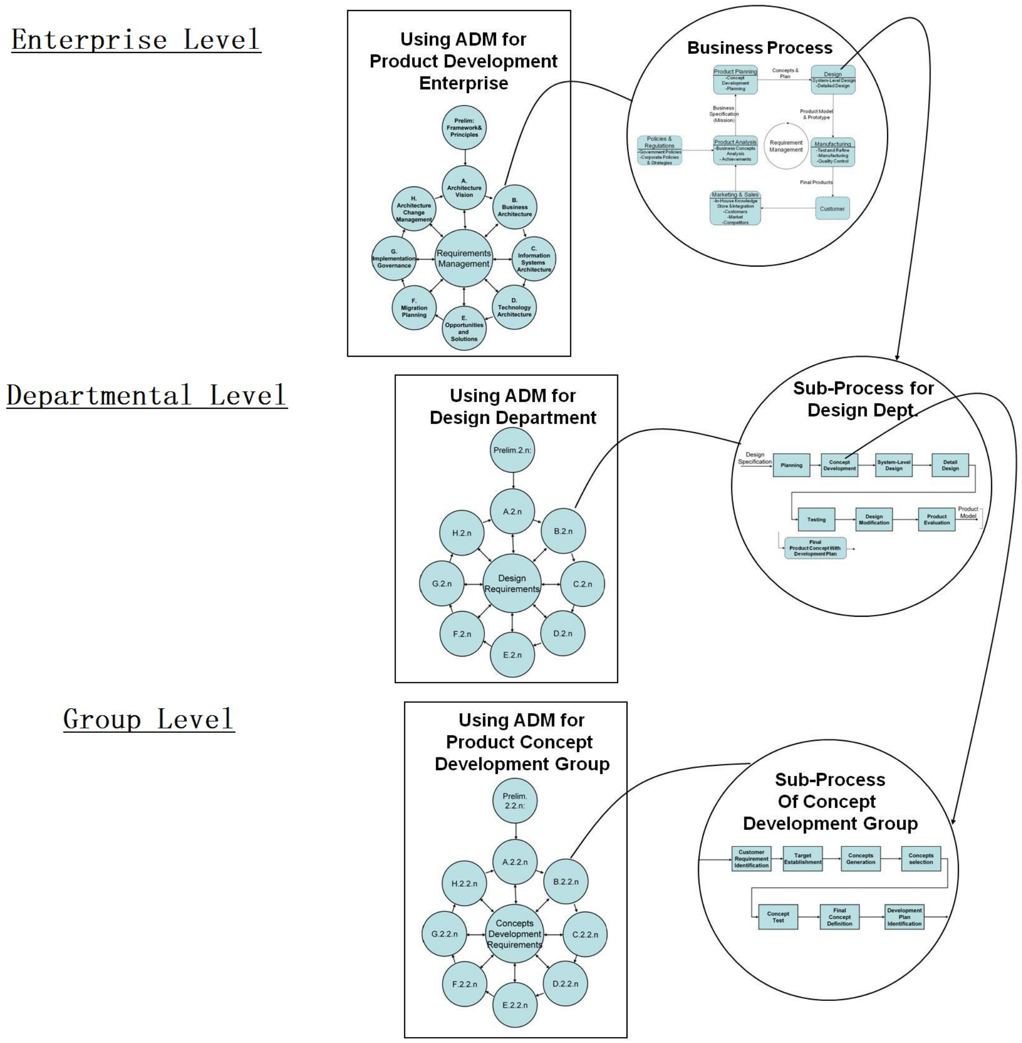


Figure 4.6 – The Different Levels of ADM Application

For example, the centre of the ADM diagram is the business needs and knowledge user's requirements. Different Departments at the Enterprise Level have some common requirements and some different requirements for knowledge management system. The common requirements are stored as Enterprise Level knowledge in the centre of the ADM diagram, whilst the details of specific requirements of different Departments will be stored at the Departmental Level (i.e., one level lower than the Enterprise Level), and the index or information about the Departmental knowledge (i.e., Meta Knowledge) will be stored in the Enterprise Level. Similarly, different Groups in the same Department have some common requirements and some different requirements. The common requirements will be stored at the Departmental level, whilst the special requirements of Groups will be stored at the Group level with index or Meta knowledge stored at the Department level.

In the literature review, a general new product development (NPD) process is introduced and discussed. In real industrial environments, some other processes are developed based on the NPD process, such as the product design and requirement process. In this research project, the main aim is to develop a KM system design methodology for the NPD enterprise. Therefore, the product knowledge flow in the NPD process is the focus of the project.

The main purpose to understand the product knowledge flow is to recognise where the knowledge exists in the NPD process and how to manage the knowledge. A NPD process normally consists of 3 main stages which are planning, design and manufacturing (refer to Figure 3.3). Before the planning stage, there is some product analysis based on the customers, market and competitors, and after manufacturing stage, the products are delivered to customers.

The Product Analysis stage gathers knowledge from 4 sources. The first source is the competitors in the current market, and usually enterprises use it for benchmarking analysis. The second source is current market for the product which is used to discover/predict new and potential market. The third one is existing customers of previous similar products of the company, i.e., some feedbacks or some new requirements of the current customers. The fourth source is the policies and regulations of the government and corporate.

The Product Analysis stage integrates the collected knowledge with the in-house product knowledge from the previous projects and some new government regulations or corporate policies. The output product knowledge is the business specification and missions. It will be input to the Product Planning stage. The Product Planning stage integrates the business specification and missions and some other factors such as budget, cost and development period, and then the development plan is produced. The product concept is then developed based on the development plan and integrated previous knowledge. After the product concept is developed, it is passed to the Design stage with the development plan and further specification configurations of the product.

The Design stage develops system configuration of the product and detail of components based on the concept, plan and detailed specifications. When the system-level design and detailed design is finished, the product drawings are transferred to the Manufacturing stage. The Manufacturing stage produces the manufacturing plans and manages the manufacturing operations. After the product is manufactured, the manufacturing knowledge is integrated and stored as the in-house product knowledge which can be used to support further product development. In this whole product knowledge flow, the product knowledge is iterative in the NPD process. The cycle guarantees that product knowledge is captured and stored in time. It also avoids product knowledge missing when the knowledge is invoked.

4.3. Applying ADM in Departmental Level Process

According to the industrial investigation, the Product Management Department is one of the five key departments in the whole product development process. It is mainly responsible for the Planning stage in the product development process. The main functions of the Product Management Department are developing product concepts and generating development plan with detailed specification. Therefore, from the management perspective, the Product Management Department is involved from the start of the business process to the end of the product concept development. Therefore, this project uses the product management department as example to show how the full ADM cycle works at department level.

Figure 4.7 shows the process model of the Product Management Department with its individual activities identified during the case study. The process modelling technique used is Design Roadmap (Park and Cutkosky, 1999). The basic elements of Design Roadmap are tasks and features. A Task (shown as a rectangle) is the primary unit of the process, which represents a function or action in the process. A Feature (shown as a rectangle with round corners) is the input and output of Tasks. The arrowed lines are used to represent the process flow and link the Tasks and Features together. The hash line represents a feedback loop.

The Activity prior to this process model is Product Analysis. Its output is the business specification and mission statement which is the starting point of this model. When the Product Management Department receives the business specification which includes the business need and mission statement, the Planning task can start. As defined in the Design Roadmap, Tasks may include lower level sub-processed based on specific needs. The output of the Planning Task is the development plan, which is the input of the next Task - Product idea Identification. In this task, there are more than one product idea identified, and one of them should be defined as the output of this task. The selected product idea is called initial product specification, as input to the next Task which is Concepts Generation. In the practical situation, there are several possible product concepts generated. Therefore, all of these product concepts need to be evaluated and selected in the next task based on various constraints. The selected one needs to be tested. If the selected concept passes the test, the final product concept will be identified. If the selected product concept does not pass the test, there is a feedback loop (dashed line in the figure) to the product concept generation task and the process is repeated, then detailed specification will be generated based on the selected concept. The output of the process is the detailed design specification. The detailed design specification will be inputted to the Design Activity in the enterprise-level business process.

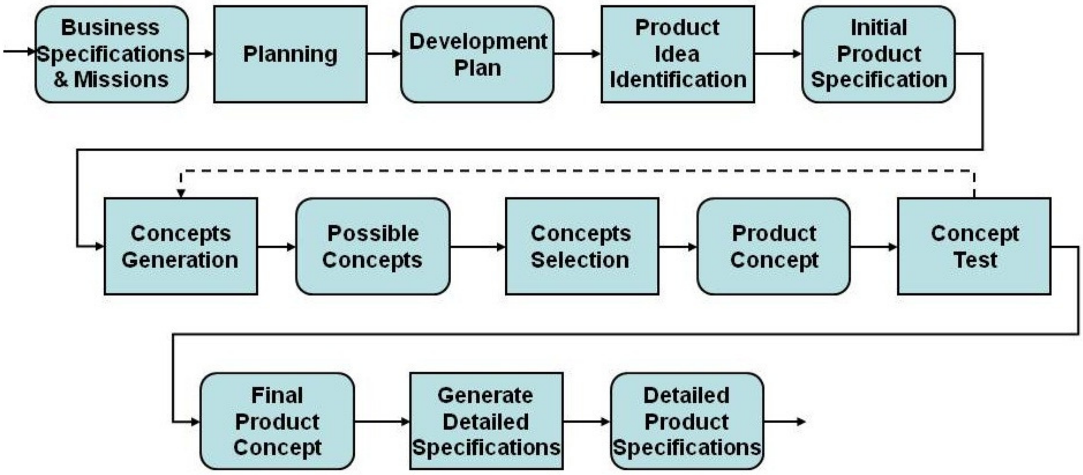

Business Specifications & Missions
Planning
Development Plan
Product Idea Identification
Initial Product Specification
Concepts Generation
Possible Concepts
Concepts Selection
Product Concept
Concept Test
Final Product Concept
Generate Detailed Specifications
Detailed Product Specifications

Figure 4.7 – the Process of the Product Planning Department as Modelled Using Design Roadmap

For example, with the car body frame, the business specification and mission is a balance between the business requirement and the business capability. It is a goal or achievement of the product, such as acquiring 3% market share through this vehicle. The product specification of the car body frame includes market segmentation and position, rules, cost, feasibility, quality, cost, performance, the entry condition of the target market and so on. Product concept is the characteristic of the product, in other words, it is the selling point for customers. The product concept of the car body frame includes information about shape of the body, NVH (noise, vibration, and harshness), safety, and so on. The final output is the detailed design specification which contains many design requirements based on the product concept, such as dimensions based on the shape of the car body frame, the ability of the car crash and rigidity based on safety. All of these factors form the design specification and then are delivered to the Design Activity. The most important knowledge at the department level is the product concept. In the collaborating company, there is a unified report template for each product concept. The product concept report is the main knowledge source to be managed. There is some supporting knowledge to develop product concept, such as the sales objective report, segmental market objective analysis, supplier research report, labour resource planning report and so on. Currently, in the collaborating company, all these kinds of supporting knowledge are managed by each department rather than managed in a common knowledge management system. This might lead to several problems, such as the lack of communication, and extra cost and time, because of misunderstanding.

According to ADM (see figure 4.4), the business process of product concept development is the basis of the Business Architecture in ADM. As in the TOGAF descriptions, each cycle has its objectives, approach, inputs and outputs. The main objective of the Business Architecture is to identify the baseline and set up an appropriate business environment including business principles, business goals and functions of the enterprise.

For the Product Management Department, one of the business goals is to identify the product concept of the car body frame. When the Business Architecture is developed, the relevant Business Architecture resources (knowledge) need to be available from the Enterprise Continuum and Architecture Capability Framework so that relevant existing models can be referred to. The supporting knowledge which is stored in the Architecture Capability Framework can be invoked when it is required.

The Enterprise Continuum provides a relationship of the supporting knowledge with its Infrastructure Integration Information Reference Model, such as potential problems with solutions and the lessons learned from the previous projects with the current relevant knowledge. The main output of the Business Architecture is a baseline for the Information Architecture including both Data Architecture and Application Architecture with constraints. This baseline contains business modelling, such as business activities and data or information exchanges in and between these business activities. These business activities form a completed business process of the company including inputs, outputs, controls and used resources. The Information Architecture is developed based on the business baseline, in order to satisfy the business requirements.

From the system design point of view, the system specification is developed based on the baseline and the relevant knowledge users' requirements. The system specification of the Information Architecture helps the collaborating company to build the Application and Data Architecture. The input of the Business Architecture is from the previous stage which is the Architecture Vision stage. The objective of the Architecture Vision is to validate the business principles and business goal. The strategy of the product needs to be clarified. For example, in the collaborating company, before developing the concept of the car body frame, the market position and strategy of the vehicle need to be established firstly. When the strategy of the vehicle is identified, the whole vehicle will be separated into different part such as the engine, body frame and balance system. Then the product concept is developed based on each part of the vehicle. The input of the Architecture Vision is from the Preliminary stage. The inputs contain the initial business strategy of the product. The necessary knowledge need to be used from the Preliminary stage which can integrate and invoke the relevant product knowledge, such as the benchmarking report of the competitors which is defined by the Marketing and Sales Department. This knowledge is also from Enterprise Continuum and Architecture Capability Framework, because the relevant previous product knowledge will need to be considered when the new product is developed. The Preliminary Stage is more like a preparation stage. One of the main objectives is to define who should be involved in and benefit from. The main function of this stage is to invoke and integrate the relevant knowledge from the Enterprise Continuum and Architecture Capability Framework as a support to the Architecture Vision stage.

As described earlier, one of the inputs of the Information System Architecture is the business baseline from the Business Architecture. The main purpose of the Information System Architecture is to develop the knowledge management system which includes Data Architecture and Application Architecture for the enterprise. It contains two steps in the approach which are design and implementation. The main purpose of the Data Architecture is design what kinds of data entities are required in the enterprise rather than concerned with the database design. In this research, this part is used to decide what kinds of knowledge that are required by the enterprise. For example, when to design and implement the Data Architecture for the Product Management Department, firstly the knowledge need to be considered is what kind of knowledge the department needs when they develop the concept of the body frame. However, the most difficult part of the KM system design is to help users find knowledge in the right position and at the right time when the knowledge is required. Enterprise Continuum and Architecture Capability Framework support this point of the KM system design. The Integrated Information Infrastructure Reference Model (III-Reference Model) of the Enterprise Continuum provides an opportunity for integrating the existing knowledge as a reference model, in order to classify knowledge and store with an appropriate sequence. Architecture Capability Framework provides a knowledge warehouse, and the user can find the relevant knowledge. The main purpose of the Application Architecture is to manage knowledge and use the knowledge to support the business functions. The Application Architecture needs to consider the Enterprise Continuum and Architecture Capability Framework as well when it is developed. Actually, the four architectures of the ADM should be considered and designed together, because the main objective of the next stage, the Technology Architecture, is to develop an architecture to form and support the basis of the following implementation work. These four architectures are the main body of the KM system. The rest of the stages in the circle are more likely the support for the implementation.

After the Technology Architecture is completed, the samples of the KM system should come out. Therefore, the main objective of the next stage, which is the Opportunities and Solutions stage, is to evaluate and identify which sample is suitable for the requirements. After the decisions are made, the output of this stage includes the details of the implementation plan and migration strategy. These plans are prioritised in the next stage which is the Migration Planning stage based on constraints such as costs and benefits for implementation. When the implementation plan is identified, the implementation plan is formulated in the next stage. Finally, the implementation can be executed. The last stage is the Architecture Change Management stage. It provides an architecture change management process when the changes are necessary. Another function of the Architecture Change Management stage is to support changes to the principles set up in the Preliminary stage.

Currently, this research focuses on the design of the KM system which involves the Preliminary, Architecture Vision, Business Architecture, Information System Architecture and Technology Architecture of the ADM circle. According to Vision 9 of TOGAF, one of the characteristics is the iteration of the ADM process. The iteration can be any number of cycles of the whole process. The iteration can also be one stage or several stages of the ADM circle. Therefore, going through the iterations from the Preliminary stage to the Technology Architecture stage can be supported by TOGAF.

In the practical situation, there are several possible product concepts generated. Therefore, all of these product concepts need to be evaluated and selected in the next task based on various constraints. The selected one needs to be tested. If the selected concept passes the test, the final product concept will be identified and broken down with details. If the selected product concept does not pass the test, there is a feedback loop (dashed line in the figure) to the product concept generation task and the process is repeated. The final output of the process is the design specification. The design specification will be inputted to the Design Activity in the enterprise-level business process.

Based on the industrial investigation and illustrations, the proposed methodology has been proved that it can be applied and implemented in the collaborative company. Although the case study does not cover the whole company, one of the key departments is proved that it can be fully implemented in the real industrial environment.

One of the main problems is that the stage or activity of the business process is not fully matched with the functions in the department or groups. For example, the Product Management Department not only contain the functions of producing the product concepts, but also contain one group which is to analyse the market and help the Sale Department to set up the budget and cost of the product development. The actual reason to allocate the department like this situation is that the product knowledge between these two departments are usually linked, therefore this structure is convenient and efficient.

4.4. Summary

In this chapter, the principles of the proposed methodology have been described, with emphasis on the improved enterprise architecture framework TOGAF. The product management department is used as an example to show the full ADM cycle works for the process at department level. This proposed methodology has several advantages. It can help companies to develop their knowledge management systems based on their own organisational structure. The interactions and logical links between enterprise levels can be managed by the ADM of TOGAF. Therefore, it provides an entire knowledge management system as a working environment. The proposed methodology considers both business requirement and knowledge user's requirement, in order to provide an opportunity for company to develop their own appropriate knowledge management system. The methodology also supports applications in the real business processes, i.e., the knowledge system designed using the proposed methodology would support real-life product development activities.

Chapter 5: Implementation of the Proposed Methodology

As discussed in the previous chapter, the requirement driven methodology consists of a guideline (in the form of a flowchart), an improved Enterprise Architecture Framework (based on TOGAF), and a "Folder-based EAF Implementation". The improved EAF has already been described and discussed in the last chapter. In this chapter, the main structure of the Folder-based EAF implementation will be described and discussed. Then the details of how to apply the developed methodology and the implemented folders will be described. The guideline which system developers can follow will be described step by step. Various links between the Folder-based EAF implementation, the guideline and ADM will be described.

5.1. The Folder-based EAF Implementation

The proposed methodology contains 3 elements (as shown in Figure 5.1): (i) a guideline which can be followed and used by users in order to develop the system specification based on their own circumstance, (ii) a Folder-based EAF implementation which is supporting the guideline and managing relevant knowledge in folders, and (iii) an improved EAF which is improved based on TOGAF. The improved EAF provides rules and principles for the developed methodology. The folder-based EAF implementation is developed based on the Improved EAF, in other words the improved EAF is implemented by the folder-based EAF implementation. The structure of folder-based EAF implementation is same as the ADM of TOGAF. It means that each circle stage of ADM can be implemented as a folder. The improved EAF has already been explained and discussed in the last chapter. In this section, how to develop the Folder-based EAF Implementation will be described and discussed.

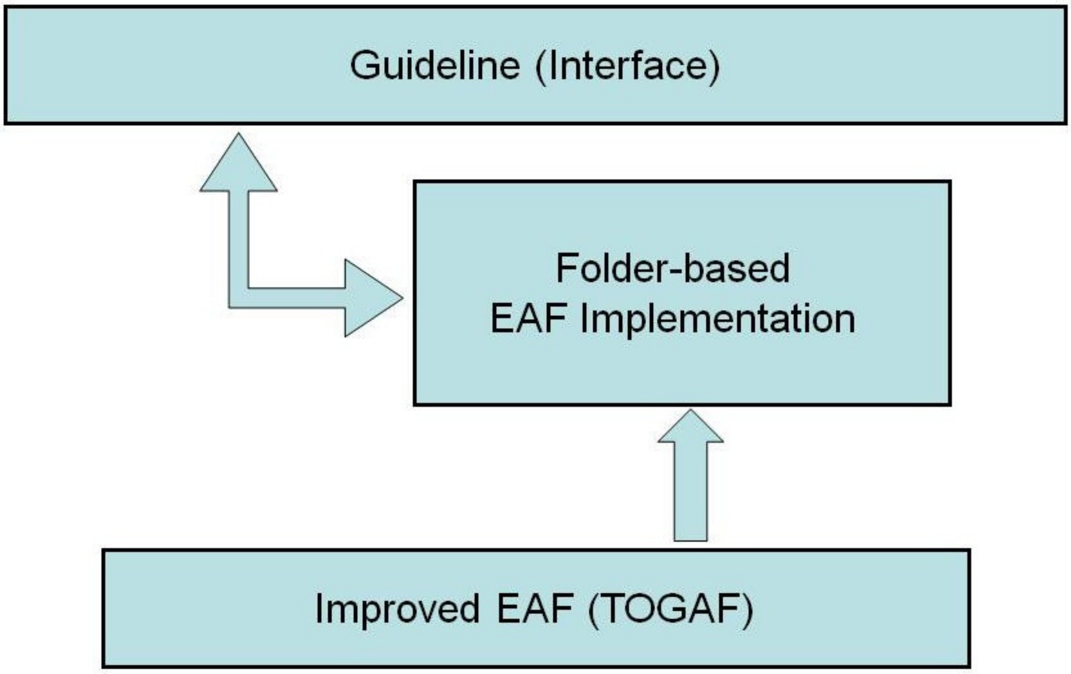


Figure 5.1 – The 3 Elements of the Proposed Methodology

The main aim of folder-based EAF implementation is to provide a tool for better using the developed methodology with a guideline for information system developers, because the Open Group does not provide a formal implementation method for TOGAF in industrial environment. This project aims to implement the improved EAF in a simple and flexible way and is independent of any specific software. Any system developer can use it with some basic TOGAF knowledge to develop the knowledge system. The relevant support knowledge can be stored and managed using the folders that are similar to Microsoft filing folders. Any format of knowledge can be contained such as figures, processes, models and so on. The Folder-based EAF implementation contains all the concepts of TOGAF such as input and output of each stage of ADM circle. If system developers are not familiar with TOGAF concepts, they just need to follow the instructions and requirements of each TOGAF component to finish their own knowledge management system. The implementation contains a set of folders and several worksheets of Microsoft Excel. The worksheets provide links to the folders which contain the supporting knowledge for system development. System developers can make use of the contained knowledge of each element of the Improved EAF. System developers can directly use the guideline to develop their own knowledge management system. If system developers want to further understand the contents of the three elements (ADM, Enterprise Continuum and Architecture Capability Framework) of the Improved EAF, they can click the corresponding icons and then the folders are opened (as shown in Figure 5.2).

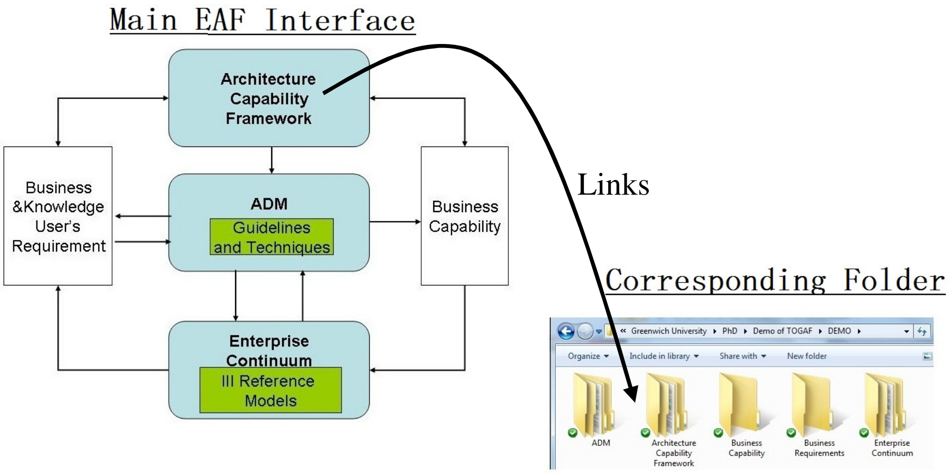


Figure 5.2 – Links between the main EAF Interface and the developed folders

When the ADM is clicked, the main iteration will come up in another Excel worksheet. System developers can click any stage of ADM to get guideline on how to develop that stage including inputs, outputs, procedures and cautions. Alternatively, system developers want to check the contents of any stage of ADM, they will also click that stage of ADM and find its contents (as shown in Figure 5.3).

Figure 5.3 represents the links of ADM stage to the corresponding folders. Each stage of ADM is represented by an individual folder. The folder contains the relevant knowledge of the corresponding stage in the ADM circle. Figure 5.3 shows two examples: folders of the Business Architecture and Requirement Management stage, which will be described below.

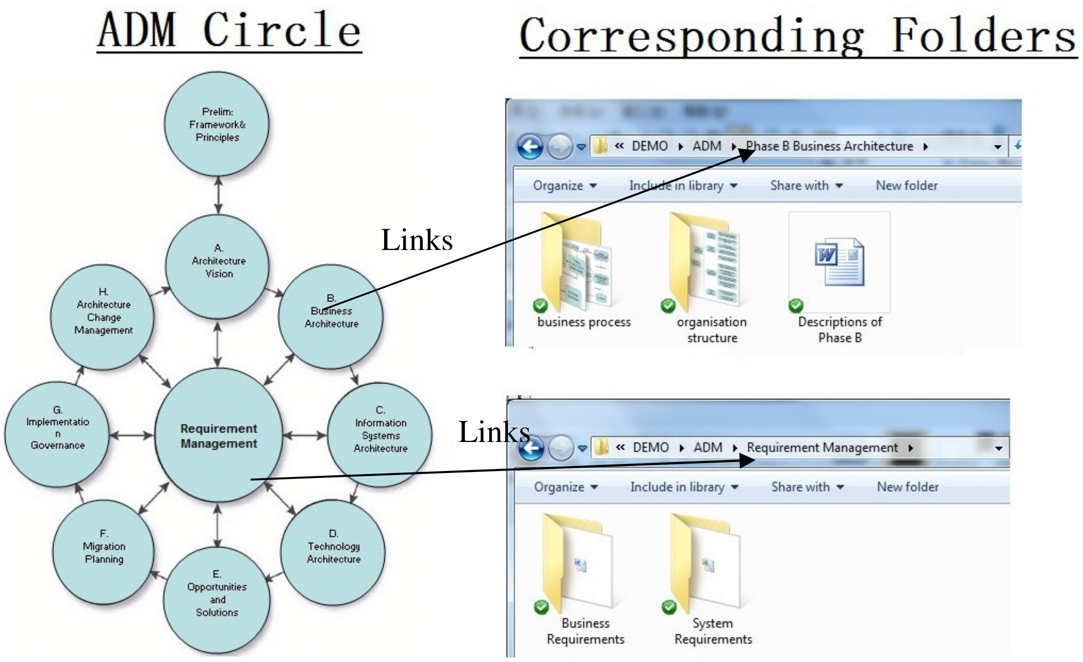

ADM Circle
Corresponding Folders
Prelim: Framework& Principles
A. Architecture Vision
B. Business Architecture
C. Information Systems Architecture
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementatio n Governance
H. Architecture Change Management
Requirement Management
Links
Links
« DEMO ▸ ADM ▸ Phase B Business Architecture ▸
Organize ▾ Include in library ▾ Share with ▾ New folder
business process
organisation structure
Descriptions of Phase B
« DEMO ▸ ADM ▸ Requirement Management ▸
Organize ▾ Include in library ▾ Share with ▾ New folder
Business Requirements
System Requirements

Figure 5.3 – Links between ADM and Corresponding Folders

The first example shown in Figure 5.3 is the Business Architecture contains 3 main elements: business process, organisation structure and descriptions. These 3 elements support system developers to build the Business Architecture. The business process and organisation structure are the supporting knowledge when the Business Architecture is developed. As discussed before, the business process is the main structure of the Business Architecture. In other words, the Business Architecture is developed based on the business process and its sub-processes. The business process folder contains the main business process and sub-processes based on each department and group. The organisation structure can help system developers associate the business process and its sub-processes with corresponding departments and groups. The description of the Business Architecture is developed based on TOGAF. It contains inputs, outputs, procedures, cautions and potential problems. When system developers develop their own Business Architecture, the description is the best way to follow and refer.

The second example is Requirement Management. As mentioned before, there is a problem for the current systems used in industry. The problem is that current systems do not give sufficient consideration on business and knowledge user's requirement. The developed methodology concerns both business requirement and knowledge user's requirement. Figure 5.4 shows information and knowledge in requirement management folder, i.e., business requirement folders and system requirements folder.

As Figure 5.4 shows, the business requirement folder mainly contains the list of business requirements as the supporting knowledge such as saving 3% of total cost of product development lifecycle, Time period reduce 5% of product development lifecycle and Obtain 3% more market share of a product. According to the improved EAF, the main approach of business requirement is contained in the Business Architecture stage. Therefore, it is the relevant knowledge to be ready for invoking when it is required. The system requirements folder contains a list of users' requirements for the KM system and relevant and supporting knowledge.

The rest of the elements of the main EAF interface are simple to follow. System developers just need to click them, and then the contents can be brought up. The contents include the relevant knowledge and the development methods.

The main advantage of the Folder-based EAF Implementation is that it provides an easy-to-use method of applying the proposed methodology. System developers can understand the perspective of their knowledge management system by just following the simple interfaces and a guideline in the form of flow chart. The relevant knowledge can be easily stored in the computer folders. It provides a flexible way to manage any format of knowledge, such as processes, models, reports and so on. The details of the "Folder-based EAF Implementation" will be provided in Appendix B.

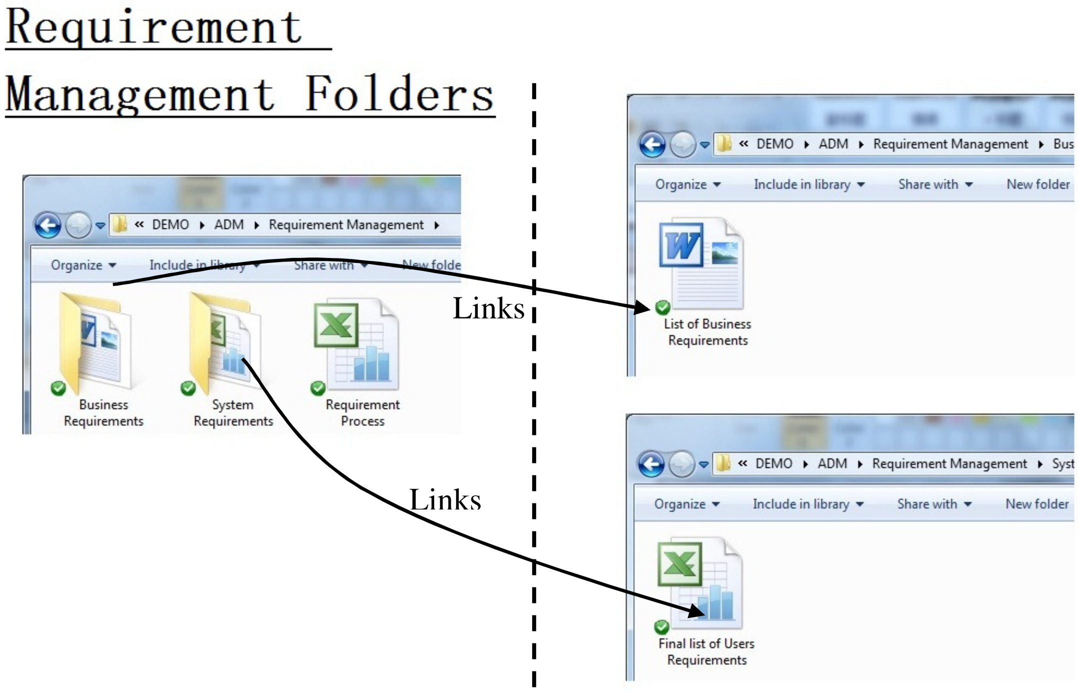

Requirement Management Folders
« DEMO ▸ ADM ▸ Requirement Management ▸
Organize
Include in library
Share with
New folder
Business Requirements
System Requirements
Requirement Process
Links
Links
« DEMO ▸ ADM ▸ Requirement Management ▸ Bus
List of Business Requirements
« DEMO ▸ ADM ▸ Requirement Management ▸ Syst
Final list of Users Requirements

Figure 5.4 – Extension of Requirement Management

5.2. Overview of the Guideline for KM System Development

As discussed before, the guideline is a part of the proposed methodology for the development of knowledge management systems (as shown in Figure 5.1). This guideline includes steps to capture and analyse enterprise knowledge and knowledge users' requirements and leads to develop system specification of knowledge management system for companies. There will be a case study of this guideline and the folder-based EAF implementation which will be described in the following Chapters.

The suite of methods and tools developed in this project, i.e., improved EAF with a guideline and a folder-based EAF implementation, allow a company to link its internal structure vertically and horizontally. For example, enterprise knowledge of the collaborating company can be divided into three levels for better managing based on its organisational structure: project level (enterprise level), functional level (department level) and task level (group level). Therefore, system developer needs to capture the business and knowledge users' requirements in three levels of the enterprise. A comprehensive and common guideline is provided, which can instruct system developers during the system development.

The aim of any system and software development project is to transfer users' requirements to system specifications. However, requirements for knowledge management systems are more complex than the user's requirements for traditional IT system. More companies find that a set of single numbers and facts in databases do not mean much, and are not enough to be used in decision making in the current business environment. That is the reason for the collaborating company to establish a knowledge management system. The users' requirements are complex in knowledge management system. Sometimes knowledge users can not clarify exactly what knowledge they require and generate. Therefore, a guideline can help system developers specify knowledge users' requirements and knowledge they require and generate. The guideline provides a process which concerns with both enterprise knowledge and knowledge users' requirements at the same time, in order to develop system specification through capturing, analysing and integrating knowledge and knowledge users' requirements.

The knowledge and users' requirements always combine and influence each other. The existing knowledge supports roles (users) to finish their daily work. At the same time, the daily work produces new knowledge. Therefore, just considering knowledge or knowledge users' requirements is not comprehensive in the knowledge system development. The users require that the knowledge system should help users to finish their daily work and manage both existing knowledge and new knowledge. However, knowledge is not the same as data and information. Most knowledge users can not clarify the knowledge during the daily work. This situation might lead to the problem that the users' requirements cannot cover all the necessary functions of knowledge management system. Therefore, the involved knowledge needs to be identified when knowledge users' requirements are captured, in order to provide a comprehensive system specification.

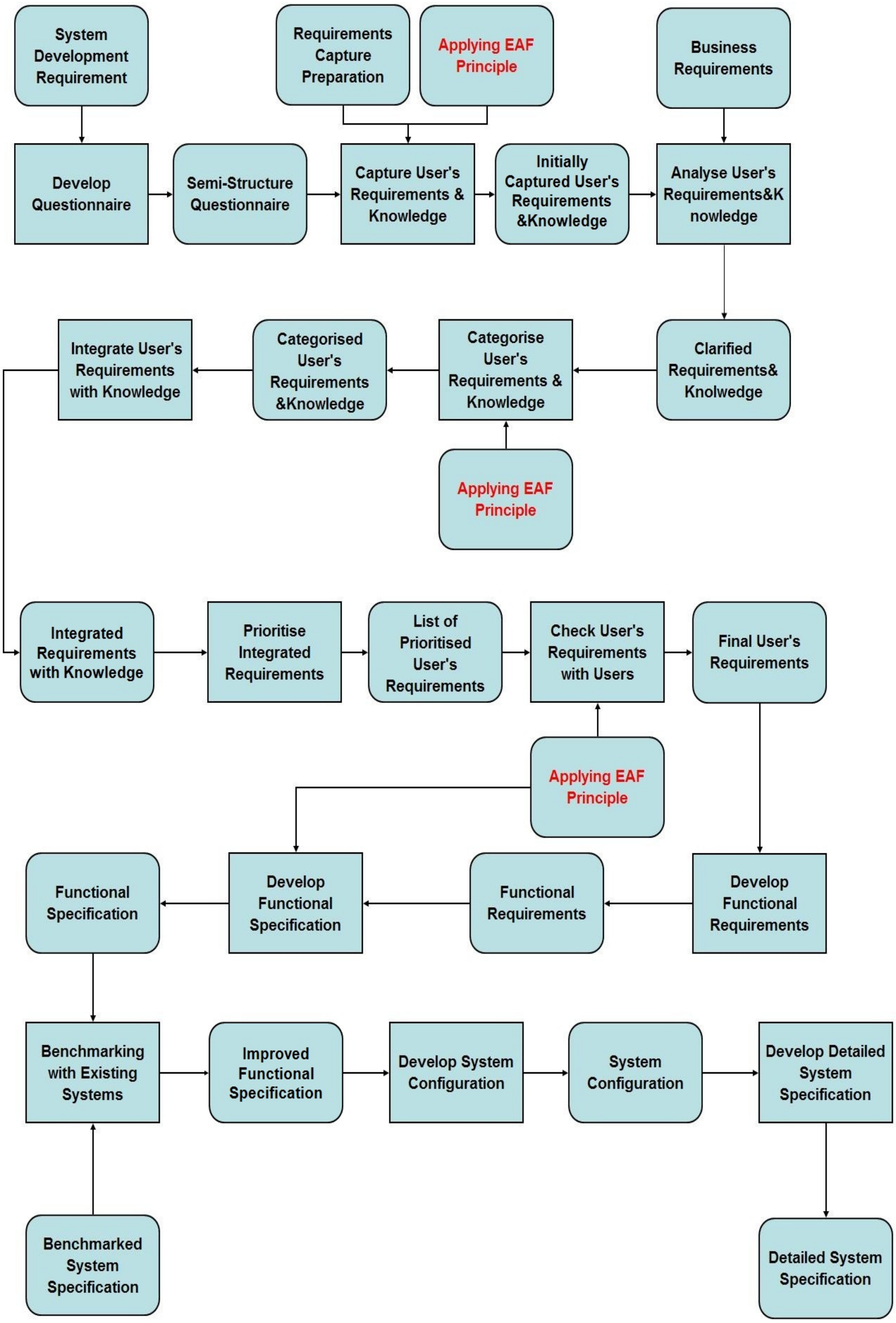

System Development Requirement
Requirements Capture Preparation
Applying EAF Principle
Business Requirements
Develop Questionnaire
Semi-Structure Questionnaire
Capture User's Requirements & Knowledge
Initially Captured User's Requirements &Knowledge
Analyse User's Requirements&Knowledge
Integrate User's Requirements with Knowledge
Categorised User's Requirements &Knowledge
Categorise User's Requirements & Knowledge
Clarified Requirements& Knolwedge
Applying EAF Principle
Integrated Requirements with Knowledge
Prioritise Integrated Requirements
List of Prioritised User's Requirements
Check User's Requirements with Users
Final User's Requirements
Applying EAF Principle
Functional Specification
Develop Functional Specification
Functional Requirements
Develop Functional Requirements
Benchmarking with Existing Systems
Improved Functional Specification
Develop System Configuration
System Configuration
Develop Detailed System Specification
Benchmarked System Specification
Detailed System Specification

Figure 5.5 – The Guideline for Knowledge Management System Development (before Coding)

In this project, most complex procedures are modelled using the Design Roadmap which is a process management technique available in the market. As Figure 5.5 shows, the requirement transferring process of this project is developed using Design Road-map. The Whole process is composed of the basic elements of Design Roadmap which are Tasks and Features. A Task (shown as a rectangle) is the primary unit of the process, which represents a function or action in the process. A Feature (shown as a rectangle with round angle) is the input and output of Tasks. The Arrowed lines are used to represent the process flow and link the Tasks and Features together. The Dashed Line represents feedback loop in the process. The flow chart is actually the KM system development process before coding (refer to the Prototyping methodology for software development).

According to the developed methodology, enterprise knowledge can be divided into 3 levels for better managing: project level (enterprise level), functional level (department level) and task level (group level). Therefore, the knowledge and users' requirements are captured based on these 3 levels. Semi-structured questionnaire interview is adopted as the capturing method. The interviewees include product manager from the enterprise level, department leaders, group leaders and members. Product manager can indicate the requirements of knowledge in enterprise level because of managing whole product development process, such as standards and principles of the company.

5.3. Detailed Explanation of the Guideline

As Figure 5.5 shows, the first task of the guideline is to develop the questionnaire. The input is requirement from a company for a knowledge management. Semi-structured questionnaire is then developed based on the system requirement which describes some business targets, such as using knowledge management system to manage the whole product development process to reduce lead times and production failure rate. The guideline also provides some samples in each task to help system developer. For example, the guideline will provide sample semi-structured questionnaire used in similar purposes. The questionnaire should contain two main parts: possible required knowledge to carry out tasks and users' requirements for KM system to support their tasks.

The user's requirements in the collaborating company are complex. The knowledge management system is generally required to manage business process, product knowledge and knowledge flow in the collaborating company. Unfortunately, in the collaborating company, there is no standard for representing the users' requirements. The initial users' requirements have to be gathered from the users directly by interviewing with semi-structured questionnaire.

The "Capture users' requirements and knowledge" task aims to gather both relevant knowledge and knowledge users' requirements in the company. There are two other inputs of this task: "requirement capture preparation" and "Applying EAF Principle". These two inputs mainly provide a development scope and a set of conditions for gathering the knowledge and knowledge users' requirements. The "requirement capture preparation" contains relevant knowledge about capturing knowledge users' requirements, e.g., interview schedule, description of interviewees and example of questionnaires. This knowledge is stored in the Resource Base, and it can be shared with relevant staff in the collaborating company. The "Applying EAF Principle" is the concept which can divide the whole enterprise into 3 different levels based on both organisational structure and product development project. This concept of "Applying EAF Principle" carries through the whole requirement transferring process. The "capture users' requirements and knowledge" task uses the principle to classify interviewees and identify kinds of knowledge and requirements needed to be captured, in other words, the semi-structured questionnaire is used to capture knowledge and knowledge users' requirements based on each individual department and group.

In the interviews, the same interviewee can provide both relevant knowledge and knowledge users' requirements. The only problem is that interviewees may not fully understand the knowledge in their daily work. Therefore, the developed methodology provides a check list of knowledge which contains all the possible knowledge during daily work. This list is developed based on real experiences and literature survey. System developers can use this list to confirm all the knowledge on the list with interviewees during the interviews, in order to capture a comprehensive list of knowledge.

The output of the "capture users' requirements and knowledge" task is a structured "initially captured requirements and knowledge". It might be developed as two different lists. These requirements and knowledge are listed based on each interviewed knowledge user at each level. However, there is no guarantee that each requirement or knowledge is directly usable based on the different backgrounds and main concerns of interviewees at different levels. They need to be analysed and integrated to avoid duplications and overlaps. The next task is "analyse users' requirements and knowledge" task. The main aim of analysing knowledge users' requirements is to point each requirement for the requirement prioritisation. Knowledge users and system developers use 100-points method to mark each requirement from 1 to 100. The mark of knowledge user is based on how important each requirement is. The mark of system developer is based on how easy to apply this requirement in the system. The final mark combines both marks from knowledge users and system developers with equal weighting.

There are also 2 main aims of analysing knowledge in this task. The first aim is to categorise the initial captured knowledge into explicit and tacit. The managing method of these two kinds of knowledge is different. As discussed before, explicit knowledge is easy to describe, such as rules and principle of new product development. Tacit knowledge is difficult to describe, even knowledge users themselves might not identify the tacit knowledge which is used and generated in their daily work, such as lessons learned and individual experiences. Tacit knowledge is important, at the same time it is easily to be ignored during daily work.

The second aim of the "analyse knowledge" task is to identify where the knowledge exists in the company. In the digital era, every company must have system to help manage the daily data and information, even some knowledge has already been managed in the system, companies still store and manage knowledge on paper, such as processes and models. Some tacit knowledge may also be stored in people's brains. Therefore, identifying the current situation of knowledge management in the company is important, because all the different formats of managing knowledge have to be converted to a format in one knowledge management system. It may lead to different converting method between different knowledge storing formats. The output of this task is “analysed requirements and knowledge”.

The next task “categorise users’ requirements and knowledge” aims to categorise the analysed requirements based on the points from both knowledge users and system developers, and then to classify all kinds of knowledge to a logical way, in order to manage knowledge in 3 levels of the enterprise structure. In this task, the requirements are no longer considered based on each interviewee. Requirements need to be viewed from system development perspective. When the boundary of each interviewee is broken, there are some obvious duplication and overlapping problems appeared, particularly at the function/department level. The duplicated requirements should be combined to one requirement. The highest mark of duplicated requirements should be adopted as the important number of the new requirement. For example, requirement A and requirement B are separately required by two different knowledge users. The marks of them are 140 and 125 from different user’s perspective. Actually, requirement A and B are the same or similar from the system development view. Therefore, requirement A and B have to be combined. The representation of the important number of the new combined requirement is 140 (140, 125), because the point of the requirement A is higher than the requirement B. The original point of each requirement is also kept in the bracket. It means that this requirement is possible to be required and used by more than one department at the same time. This task is more like a preparation for requirement prioritisation.

Moreover, the new requirement might be changed to upper level in the organisational structure. For example, there is a requirement which is needed by more than one department. Unfortunately, department leaders do not know that there are some other departments that need this system function as they have not communicated about this. The duplicated requirement is generated in this circumstance. During the interviews, this kind of requirements can be integrated and moved to the upper level which is project/enterprise level. The output of this task is “categorised requirements”. All of them should not be duplicated and overlapped.

The classification of knowledge in this task contains 3 different categories based on the industrial investigations and real situation in the collaborating company. There are 3 categories of knowledge existing in the collaborating company. The first category is knowledge existing and shared in all 3 levels, such as the product concept report. This category of knowledge is popular and is required by most departments and groups, even required by product manager and the board. The second category is the knowledge specifically existing in one single level, such as a management discipline of a group. This category of knowledge is not common or shareable in the collaborating company, and its pertinence is just focused on one group or department. The third category of knowledge is that the knowledge in the higher level is shared by lower levels, such as the business process and product design model. This category of knowledge needs to be considered and managed vertically and horizontally. This category of knowledge is the most complex knowledge in the collaborating company. Therefore, it needs to be managed based on individuals, and then the knowledge based on the individual can be managed systematically under the organisational structure. More details and examples will be discussed in Section 6.4.

In the current stage, knowledge and knowledge users' requirements are still contained in two different lists. In the next task, knowledge and knowledge users' requirements are integrated as one list, in order to product the final list of knowledge users' requirements. The list of knowledge enhances list of knowledge users' requirements.

Some necessary system functions are not raised by the users directly in the list of users' requirements. For example, from product development point of view, the knowledge needed includes the development process, in-house previous related product knowledge and their customers feedback, the problems encountered during development process and the methods used to fix the problems, lessons learned from the previous projects, previous product models, reports for potential benefits, experts experience with the new product, meeting minutes in both previous projects and current project, potential market and customer investigation, records of external suppliers, contractors and partners. In all these kinds of knowledge, some knowledge might have already been represented with some formats, such as digitalised product models and standardised product development process. The formatted knowledge has already been recognised to require some related system functions to manage it, such as the geometric model in the knowledge management system. Some knowledge might not have been represented but exist somewhere in the enterprise or human heads. The system functions to manage this kind of knowledge should be suggested by system developer based on the list of knowledge if it is not recognised by knowledge users. For example, during the product development process, there are some lessons learned previously. If these lessons are not recorded and managed systematically, it may be repeated again when the same problem is faced. It may lead to extra and unnecessary cost in the product development. Therefore, a system function is required to record and manage the lessons learned during previous product development process.

Therefore, the “integrate users’ requirements with knowledge” task aims to provide a comprehensive list of knowledge users’ requirements for developing system specification. The output of this task is “integrated requirements with knowledge”, and then it is inputted into the next task.

The “prioritise categorised requirements” task aims to divide integrated requirements into 3 categories: necessary, useful and wanted, and all requirements are ranked based on the prioritised point. As discussed before, requirement prioritisation is aimed to help enterprise to save unnecessary cost and system waste. The necessary requirement is essential requirements which must be included in the system, useful requirements which could be reduced if the system or project is out of the boundaries, and wanted requirements which make the system more desirable. In this project, the criteria and boundaries are discussed and identified by knowledge users and system developers. The point greater than and equal to 120 is the necessary requirement. The point between 100 and 120 inclusive is the useful requirement. The point below 100 is the wanted requirement. It is obvious in the ranked requirements that the top requirements with the high priority in the list are the necessary requirements. The useful requirements are in the middle, and the wanted requirements are at the bottom. The “list of the prioritised users’ requirements” is founded formally as the output of the task “prioritise categorised requirements”. This output is also inputted to the next task which is “check users’ requirements with users”. The task will double check and confirm the list of prioritised users’ requirements with users before moving onto the next stage. If there are any requirements which are not satisfied with users, these requirements will be modified until all the requirements are satisfied by knowledge users. The output of this task is the “final users’ requirements”. This output is used to developed system specifications.

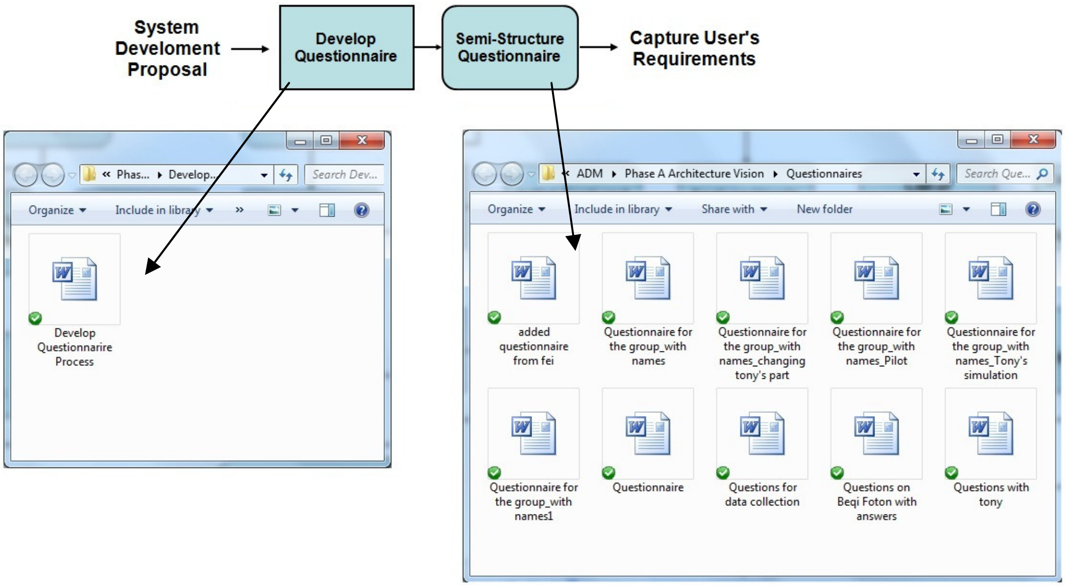
According to modern system or software development techniques, most of system specifications are developed based on the users’ requirements. The final users’ requirements are converted to the functional requirements of the system directly. It is one to one conversion. When the functional requirements are developed, it can be found that there are many duplications and overlaps between functional requirements. Therefore, the next step is the development of the functional specification, in order to remove all duplication and overlaps and combine functional requirements.

The functional specification is still not used to develop the system specification directly, because system specifications are not only from the users’ requirements, but also are developed based on other relevant or reference systems. Therefore, the functional specifications need to be improved through benchmarking with existing systems, in order to improve functional specifications comprehensively. The next step is the development of the system configuration based on the improved functional specifications. As discussed before, this research is followed the Prototyping methodology of the system development. The developed system specification is used to develop system configuration which provides a perspective for easily understanding the knowledge management system, and then system configuration can be used to developed system specifications which provide a specific system development details to the programmers. The system specification is the final output of this project. There will be more details and discussions in the following chapters which illustrate a case study for this process. The details of all inputs and outputs are provided in Appendix B.

5.4. *Folder Implementation to Support the Guideline*

Since there is a large amount of relevant knowledge to be considered, clarifying the various links in the guideline is important i.e., links to the Folder-based EAF implementation described in section 5.1, and the links to the implemented folders to support the guideline.

Firstly, each element, including tasks and features, needs to be modelled as a single folder. There are two main elements of the guideline based on Design Roadmap technique. The task represents activity of the process, and the feature represents input and output of the process. Therefore, folders for these two different elements need to be developed separately. Figure 5.6 shows an example of links between the guideline and its relevant folders (also refer to Appendix B).



System Develoment Proposal
Develop Questionnaire
Semi-Structure Questionnaire
Capture User's Requirements
« Phas... ▸ Develop...
Search Dev...
Organize ▾
Include in library ▾
»
Develop Questionnarire Process
« ADM ▸ Phase A Architecture Vision ▸ Questionnaires
Search Que...
Organize ▾
Include in library ▾
Share with ▾
New folder
added questionnaire from fei
Questionnaire for the group_with names
Questionnaire for the group_with names_changing tony's part
Questionnaire for the group_with names_Pilot
Questionnaire for the group_with names_Tony's simulation
Questionnaire for the group_with names1
Questionnaire
Questions for data collection
Questions on Beqi Foton with answers
Questions with tony

Figure 5.6 – Example of the Folders of Tasks and Features

All folders of the tasks in the guideline contain processes. This process aims to show the procedure to the system developer who used this process. The process normally contains several steps, and the cautions or potential problems might happen during the developing process. Comparing with tasks, the folder of the feature contains relevant knowledge. This knowledge is not only the supporting knowledge for the next stage of the guideline, but also the generated knowledge from the last stage. This knowledge can be any format, such as report, models, processes, journals and meeting minutes. All of these folders exist individually in the Folder-based EAF implementation.

All relevant knowledge is managed in the ADM based on the circle stage definition of the ADM. The knowledge in supporting stage of the guideline is mainly managed in the Architecture Vision circle stage of ADM, such as the tasks and features about questionnaires, proposal and the Applying EAF Principle. The tasks of the requirement capturing branch are managed in the Application Architecture. The features of requirement capturing branch are managed in the system requirement management. The system development stage is also managed in the Application Architecture. The tasks and features of knowledge identifying branch are managed in the Data Architecture. Both Application Architecture and Data Architecture are contained in the Phase C Information Architecture. The periodical results and final results are managed in the Opportunities and Solutions such as suggested requirements, system design specifications and implemented functions of the system specifications. All of these identifications and definitions are developed based on TOGAF. System developers do not need to understand all of these links. They just need to follow the guideline and invoke the supporting knowledge and guideline of each task.

5.5. Summary

The guideline is developed as an appropriate way to apply the developed methodology. This chapter mainly discussed the Folder-based EAF implementation from three aspects, i.e., the main structure of the interface, the links of the Folder-based EAF implementation, and the flowchart of the guideline.

Chapter 6: Capturing and Analysing Users' Requirements Using the Developed Methodology

In the previous chapters, the basic theory and principles of the developed methodology have been discussed. In the collaborating company, engineers and managers usually cannot formalise the knowledge they really needed and the knowledge they produced. This situation leads to knowledge loss during the daily routine work. This chapter will describe how to capture and analyse the knowledge and knowledge users' requirements in the product development process of the collaborating company. The guideline will be followed until the final list of users' requirements is generated. The rest of the processes in the guideline will be described in the following chapters.

As described previously, the organisational structure has a hierarchical tree structure. The top level is the enterprise level. The second level is the department level consisting of 5 departments involved in the product development process. The bottom level is functional groups belonging to their parent departments. To simplify the description of the case study, at the department level, only the Product Management Department and its groups will be used to illustrate how the developed methodology works.

6.1. Method for Capturing Knowledge and Knowledge Users' Requirements

As discussed in the literature review and industrial investigation, this project carried out interviews with semi-structured questionnaires to capture data from the collaborating company including the knowledge which exists in the product development process, business requirements of the company and knowledge users' requirements in the product development process. As mentioned before, the product development part of the collaborating company is defined as an individual "enterprise" in this project. The knowledge users are the roles which are involved in the enterprise. In the current situation, knowledge, particularly tacit knowledge exists in the heads employees of in the collaborating company. Therefore, the face-to-face interview with semi-structured questionnaires is the most appropriate method to capture the knowledge in the company.

The process starts from the board of directors of the company. The board concerns that the internal knowledge is grown faster and faster in recent years. The unmanaged knowledge leads to several problems and extra costs to the company. Therefore, the board proposes to develop a knowledge management system as business requirements. The proposal contains the targets and problems which the board has to concern. For example, the production failure rate is still high because of unmanaged product knowledge. The board seeks to use knowledge management system to reduce the failure rate by 20%. The knowledge management system is driven by these business requirements.

The semi-structured questionnaires are used to capture both knowledge and knowledge users' requirements from interviewees. The car-body framework is selected as the example component. The task is to identify what knowledge is required when the car-body framework is designed. Other relevant information of the design of car-body framework is also required, such as structure and process of the development. The required information and knowledge about the example component is given as following:

- ➢ Fully understanding of the structure of the system
- ➢ Understanding of design process
- ➢ Technical requirements

- Conditions and constraints
- Functions of the car body framework
- Other details such as number of components, suppliers, drawings, roles and departments, groups and individuals involved
- The details of system-level design
- What kinds of supports needed (not only the information system support, but also functional supports from other departments)
- The development plan
- Concept development applied or implemented

From the list of required information and knowledge, the main concern is what knowledge is required when the example product is designed. After setting up the list of required information/knowledge, the open-question questionnaire can be developed. Each required information/knowledge on the list can be developed at least one open question. The examples of the questionnaires are shown as following:

Example Questionnaires for Designers and Project Mangers

Name:

Date:

Position:

- Could you describe the details of the structure of the car body?
- Could you describe the design process of this car body?
- When you start to design the car body, what kinds of conditions and requirements you need to consider in the design process?
- What functions are necessary to be performed by in the car body? Are there any functions you have to give up when you develop the car body? Why?
- What kind of information/knowledge you need to consider and involve in the car body design?
- What is your input and output for designing?
- In the whole design process of the car body, what kinds of roles, department and suppliers need to be involved?
- When you develop the product, what kinds of supports you need?
- How do you plan the process of the product development?
- How do you apply and implement product concept development in this system?

In the questionnaires, most questions are open and straight. The advantage is that the semi-structured questionnaires provide an opportunity for interviewees to talk as much as they want. Another advantage of the semi-structures questionnaires is that interviewer can ask further questions based on the questionnaires when the interviewer thinks that the answer is not satisfying.

Another example questionnaire is to identify both knowledge and knowledge users' requirements for the knowledge management system. Knowledge and users' requirements are mixed together in the collaborating company. The interviewing objective is the same from the system development point of view. Therefore, one set of questionnaires can identify both knowledge and knowledge users' requirement. The collaborating company can be divided into 3 levels: enterprise level, department level and group level based on EAF Principle. The questions are set up with common features in all 3 levels. The examples of the questionnaires are given below:

Example Questionnaires about Knowledge Systems

Name:

Date:

Position:

- What knowledge do you need and generate in your daily work? An example please? (Need to clarify explicit and tacit knowledge, structured, semi-structured and unstructured knowledge. If interviewee cannot figure out all of them, suggestions need to be made in the interviews).
- Do you have any PLM or ERP systems in use, if you have, could you please describe it? If not, do you have any similar systems or applications in your company?
- What knowledge is needed in this system from a knowledge system user's point of view? (When talking about the KM needs, 3 levels need to ask organisation requirements, product requirements and KM system requirements).
- How to manage knowledge or information in current situation of the example system, any limits in the current situation, any more requirements need to be considered in KM system?
- How do you store your knowledge and information? Diagram, text, paper and pen, computer files?
- How do you communicate and share knowledge with other departments or roles?

To sum up, the face-to-face interview with semi-structured questionnaires provides a flexible environment for both interviewer and interviewees. Interviewer can make it clear exactly what data and information required from the interviewees. The results of the questionnaire will be discussed in further sections in this chapter.

6.2. Knowledge and Requirements Capturing and Analysis in the Collaborating Company

In this section, the tasks about capturing and analysing knowledge and users' requirements are discussed using examples in the collaborating company. The illustrative example is further explained in Figure 4.5. According to EAF Principle, knowledge and knowledge users' requirements have to be captured at 3 levels: enterprise level, department level and group level. The enterprise level will be firstly described and discussed. The Product Management Department and its two affiliated groups will be explained as examples in the department level and group level.

In the collaborating company, an interviewee from each level is selected, normally the manager of that level or a key expert of the level, such as product manager or project manager in enterprise level, every department leader in the department level, and group leaders of the group level. The reason for interviewing them is that they understand all functions of their departments, knowledge involved and knowledge flow. As knowledge users, the knowledge they required to manage is clear and easy to identify.

During the interviews there is a serious problem for capturing knowledge, i.e., knowledge exists anywhere in the collaborating company without any structural order. Some knowledge may not have been represented but exists somewhere in the enterprise or human heads. Knowledge is intensively used in the new product development process. For example, from product development point of view, the knowledge needed includes the development process, in-house previous related product knowledge and their customers feedback, the problems encountered during development process and the methods used to fix the problems, lessons learned from previous projects, previous product models, reports for potential benefits, experts experience with the new product, meeting minutes in both previous projects and current project, potential market and customer investigation, records of external suppliers, contractors and partners. In all these types of knowledge, some knowledge may have already been represented with some formats in the collaborating company. For example, the Design Department has its own system to store digitalised product models and standardised product development process. This individual system is not shared with other departments or groups, even its higher level cannot access this system. Other knowledge, especially tacit knowledge is not managed by any system.

According to the guideline, the questionnaires are used to capturing knowledge and users' requirements from the knowledge users (as shown in Figure 6.1). However, most knowledge users can not clearly specify all the knowledge they used and generated. Therefore, a check list of suggested knowledge is needed in the interviews. This list contains all the knowledge which is listed before. If the interviewees do not mention some knowledge in the list, the interviewer would remind them initially in the interviews, in order to ensure all the knowledge is checked with knowledge users.

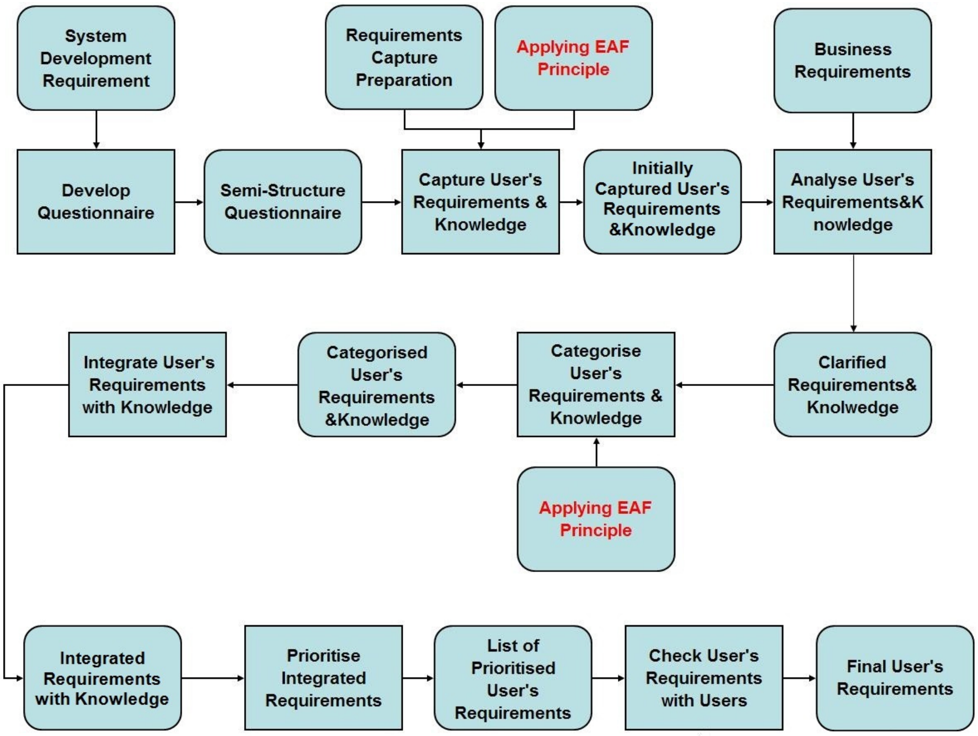
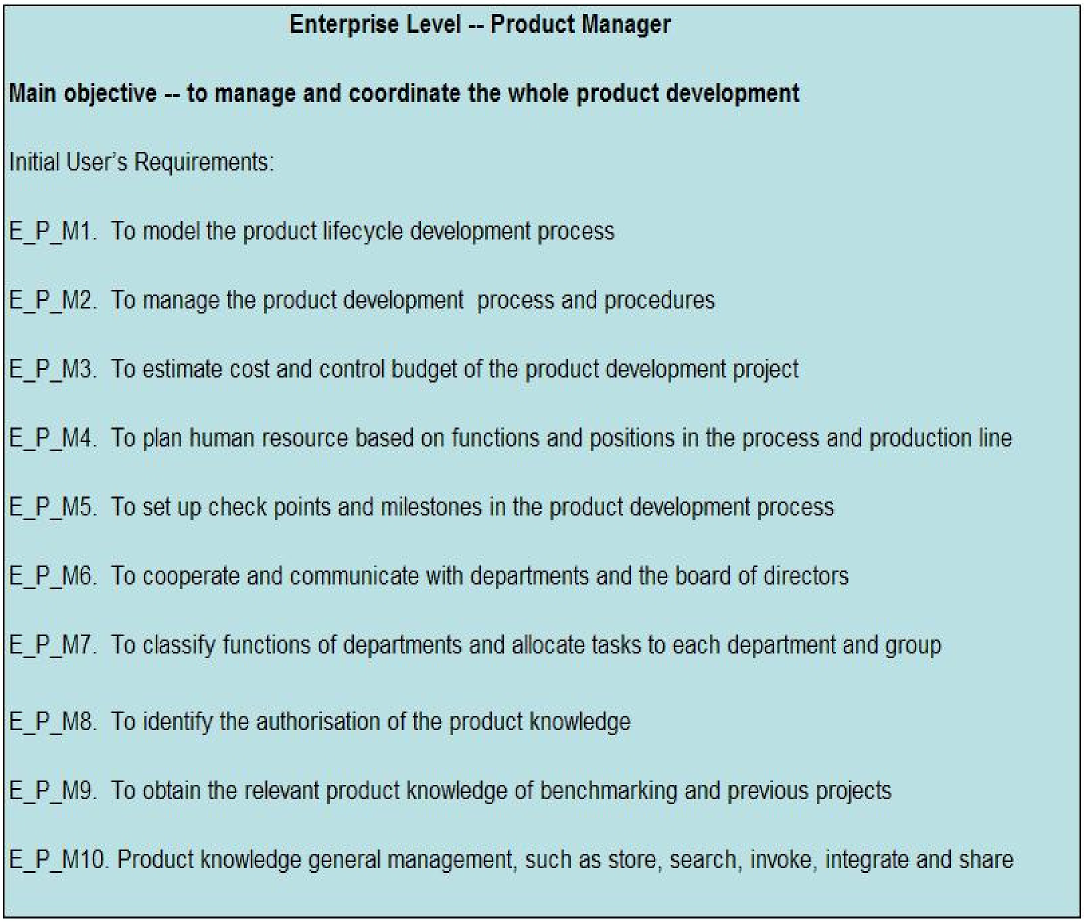

System Development Requirement
Requirements Capture Preparation
Applying EAF Principle
Business Requirements
Develop Questionnaire
Semi-Structure Questionnaire
Capture User's Requirements & Knowledge
Initially Captured User's Requirements &Knowledge
Analyse User's Requirements&Knowledge
Integrate User's Requirements with Knowledge
Categorised User's Requirements &Knowledge
Categorise User's Requirements & Knowledge
Clarified Requirements& Knolwedge
Applying EAF Principle
Integrated Requirements with Knowledge
Prioritise Integrated Requirements
List of Prioritised User's Requirements
Check User's Requirements with Users
Final User's Requirements

Figure 6.1 – Process of Developing Final Users' Requirements (Part of Figure 5.5)

Figure 6.2 shows the initially captured knowledge users' requirements at the enterprise level as an example. The other two levels are similar to the enterprise level. Each requirement statement has an ID. From the list of requirements initially captured, most requirements are not easily understood by system developers. Therefore, the requirements need to be transferred to the language which can be understood by system developers.

In the initial knowledge user's requirements, it is easily found that there are some duplicated requirements. In this project, the duplication problems are more serious, because of the 3 levels structure and lack of communication within/between levels. For example, some departments have the same requirements as the enterprise level, such as plan of human resource. This kind of requirements will be combined in the "Categorise User's Requirements and knowledge" task.



Enterprise Level -- Product Manager

Main objective -- to manage and coordinate the whole product development

Initial User's Requirements:

E_P_M1. To model the product lifecycle development process

E_P_M2. To manage the product development process and procedures

E_P_M3. To estimate cost and control budget of the product development project

E_P_M4. To plan human resource based on functions and positions in the process and production line

E_P_M5. To set up check points and milestones in the product development process

E_P_M6. To cooperate and communicate with departments and the board of directors

E_P_M7. To classify functions of departments and allocate tasks to each department and group

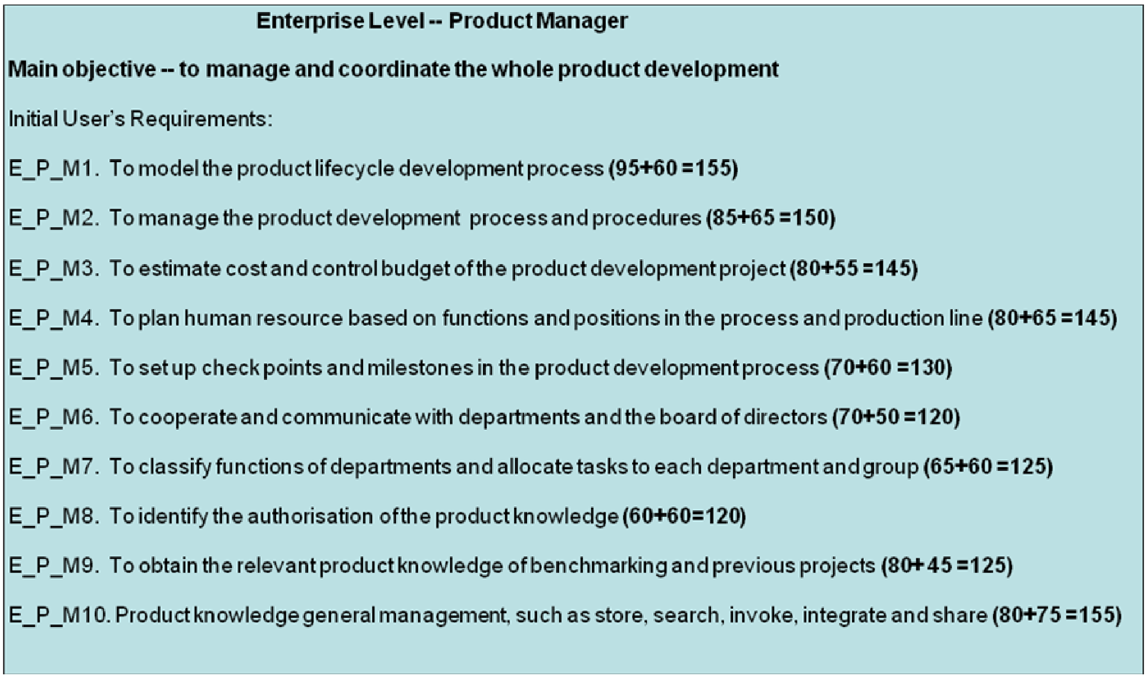
E_P_M8. To identify the authorisation of the product knowledge

E_P_M9. To obtain the relevant product knowledge of benchmarking and previous projects

E_P_M10. Product knowledge general management, such as store, search, invoke, integrate and share

Figure 6.2 – An Example of Initially Captured Knowledge User's Requirements

When the requirements are initially captured, all the requirements need to be analysed. The main aim of analysing users' requirements is preparing the requirement prioritisation. Knowledge users and system developers use 100-Points method to mark each requirement from 1 to 100, in order to indicate which requirement is concerned as high priority. The identifying standard of the knowledge user is to measure how important about this requirement based on each role. The identified standard of the system developers is the complexity of the requirement in the system development. For example, in Figure 6.3, the requirement E_P_M9 contains 2 different points at the end of the description in the bracket. The first point "80" represents the point from the knowledge user. This requirement is related to the main business process which is the new product development process in the collaborating company. The product manager can save time through accessing relevant previous product knowledge. Therefore, product manager measure this requirement as important as "80". The second point "45" is identified by system developers. To implement this requirement needs to involve many functions and many layers of the system. The system developers define the difficulty of this requirement as average and "45" is the point marked by the system developers.



Enterprise Level -- Product Manager

Main objective -- to manage and coordinate the whole product development

Initial User's Requirements:

E_P_M1. To model the product lifecycle development process **(95+60 =155)**

E_P_M2. To manage the product development process and procedures **(85+65 =150)**

E_P_M3. To estimate cost and control budget of the product development project **(80+55 =145)**

E_P_M4. To plan human resource based on functions and positions in the process and production line **(80+65 =145)**

E_P_M5. To set up check points and milestones in the product development process **(70+60 =130)**

E_P_M6. To cooperate and communicate with departments and the board of directors **(70+50 =120)**

E_P_M7. To classify functions of departments and allocate tasks to each department and group **(65+60 =125)**

E_P_M8. To identify the authorisation of the product knowledge **(60+60=120)**

E_P_M9. To obtain the relevant product knowledge of benchmarking and previous projects **(80+45 =125)**

E_P_M10. Product knowledge general management, such as store, search, invoke, integrate and share **(80+75 =155)**

Figure 6.3 – An Example of Clarified User's Requirements

The knowledge analysis is more complex than users' requirements analysis. Knowledge analysis needs to consider all the 3 levels of organisational structure. As described previously, the enterprise level is the top level of the organisational structure in the collaborating company. The main purpose of this level is that it provides a perspective of the whole product development for the board of directors of the company, in order to help the board make the right decision during the product development procedure. According to the industrial investigation, the product manager manages and controls the whole product development process in the collaborating company. Product manager is a member of the board. Product manger understands both business achievements and product development. This is the main advantage of the product manager. In this project, several product managers are interviewed to gather and identify the common knowledge which needs to be managed in the enterprise level.

According to the interviews, the most important concern of each product manager is to control and manage product development process in a certain circumstance. This concern may include a lot of different knowledge, such as personal experiences of product manager, lessons learned from previous projects, in-house products knowledge including reviews and expertise experiences, principles and rules in the collaborating company, suppliers' and contractors' knowledge. A summarised list of knowledge which is based on interviews of product managers is shown as following:

- Experiences of product managers (Tacit Knowledge)
- Organisational policies and rules (Explicit Knowledge)
- Expertise and skills(Tacit Knowledge)
- In-house product reviews and achievements, some specific knowledge about what this product can bring and why (Explicit Knowledge and Tacit Knowledge)
- Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners (Explicit Knowledge and Tacit Knowledge)
- Board decisions which are based on the existing in-house product knowledge (Tacit Knowledge)
- Enterprise level meeting minutes and reports, video and audio (Explicit Knowledge and Tacit Knowledge)
- Lessons learned in enterprise level (Tacit Knowledge)
- Benefit indication what potential benefit for a new product: technique and marketing (Explicit Knowledge and Tacit Knowledge)

- Problems address and possible solutions (Tacit Knowledge)
- Standards (Explicit Knowledge)
- Processes and relevant simulations (Explicit Knowledge & Tacit Knowledge)
- Customer feedback and investigation of both existing and potential market (Explicit Knowledge)
- Product models (Explicit Knowledge)
- Policy analysis and results (Explicit Knowledge)

There are 2 aims of knowledge analysis. The first aim is to divide the captured knowledge into 2 categories: explicit knowledge and tacit knowledge. For example, in the list, the standardisations and policy are explicit knowledge. The product manager experiences and lessons learned are tacit knowledge. The current place of storing knowledge is complicated in the enterprise level. Some knowledge has already been stored or managed currently with some formats. The managed knowledge is usually explicit knowledge, such as the standardisation of 6-levels (as shown in Appendix C) product development in the collaborating company. The 6-levels product development standardisation defines the product development in the 6 different levels based on its complexity and the roles involved. The 6-levels product development standardisation is one of the basic standardisations in the collaborating company. It is managed in the common system which can be accessed by any employees in the company. Some employees are very familiar with the standardisation. In this circumstance, the knowledge is managed in their brain directly.

Various tacit knowledge exists in the enterprise level, such as the experiences of product manager and lessons learned from previous projects. In the collaborating company, most knowledge users do not realise what the tacit knowledge is, and they do not recognise the tacit knowledge like personal experiences can be managed in knowledge management system. Therefore, the management of tacit knowledge is none in the collaborating company. One method for managing tacit knowledge is to transfer tacit knowledge into explicit knowledge. For example, in the collaborating company, an experienced product manager normally finished several product development projects successfully before. Therefore, in the product development with the experienced product managers, some decisions are easily made, such as the estimation of human resource and the perspective of production line. It will avoid some unnecessary risks and potential extra cost during the process. The experienced product manager might also predict some potential problems based on his/her experiences. Therefore, the experiences of the product manager are a valuable asset in the company. However, a big problem in current industrial environment is "brain drain". Many employees in middle level, even in high level flow to competitors or other employers. Therefore, how to keep the knowledge inside the enterprise is very important. Knowledge management system should allow every knowledge user to store the opinion and experiences of the daily work with some formats like text, video and audio in every day. Knowledge management system can manage the stored knowledge inside of system, and it can be accessed by other authorised users. This method can store experiences passively. Other ways to manage tacit knowledge is to keep meta-information about who has what knowledge so that when needed, they can be contented.

There are some examples described to help understand the knowledge situation in the collaborating company. The first example to be described is the organisational policies and rules which are typical explicit knowledge and mature in the collaborating company. One of the typical examples is the 6-levels product development principle. The 6-levels product development principle classifies the product development into 6 different levels based on the complexity of product development. Level 6 is the most complex, and level 1 is the most simple. Level 6 is normally a complete new product development, such as a new track development. Level 1 is to make some small changes on the components of existing products. The product manager mainly controls and manages level 6 and 5. High level product development needs the multi-departments collaboration. The middle level is to make some small changes based on a whole product development. The design department is mainly responsible for making the changes. The low level product development can be finished by one group or department. This principle is one of the principles and rules in the collaborating company. However, most interviewees ignored it as a knowledge user's requirement in the interview, because it is too obvious to forget clarifying it. This principle is definitely a knowledge needed to be managed in the knowledge management system.

One example of the tacit knowledge is the lessons learned. This tacit knowledge is one of the most forgettable knowledge which interviewees did not mention in the interviews. Lessons learned are also one of the most widely existing knowledge of each role in every level. Each role has some lessons learned during the daily work. These lessons learned ask role to avoid the situation in further work. For example, there was a serious unsuccessful product development in the past few years. The main reason to shut down the project is misunderstanding between the design department and the procurement department. It led to a core component that cannot be delivered on time. Finally, the delay time is too long to shut down the project. This lesson is painful, and it led to a high cost and time waste. Therefore, this situation has to be avoided in future projects. If this knowledge was not managed and recorded in the knowledge management system, the same or similar situations may happen again. However, most of interviewees do not realise the importance of this category of knowledge that needs to be managed. This is a typical gap between knowledge which needs to be managed and knowledge users suggested requirements.

According to the organisational structure of the collaborating company in the industrial investigation, there are 5 main departments involved in the product development process: the Strategy Development Department, the Product Management Department, the Product Design Department, the Product Manufacturing Department and the Marketing and Sales Department. The five departments are responsible for the main stages of the product development process.

In the department level, the Product Management Department is explained as an example to explain the situation of knowledge in the department. The procedure of capturing and analysing knowledge will be followed. The differences of knowledge between department level and enterprise level will be clarified.

The main responsibility of the Product Management Department is to plan the whole product development process including product concept development, product line design and development, integration of internal resources and collaboration of suppliers and partners. The main interviewee of the Product Management Department is the head of the department. Some other interviewees are also included such as some expertise and assistants. The knowledge to be captured is listed below:

- Experiences of staff of this level, such as department leader, key staff of each department function, e.g., main designer of car body framework (Tacit Knowledge)
- Experiences and expertises (Tacit Knowledge)

- Lessons learned (Tacit Knowledge)
- Communication records between departments (Explicit Knowledge and Tacit Knowledge)
- Problems addressed and possible solutions (Explicit Knowledge and Tacit Knowledge)
- Key department functions' explanations and key principle and rules (Explicit Knowledge)
- Department level meetings minutes and reports, video and audio (Explicit Knowledge and Tacit Knowledge)
- Relevant process and simulations (Explicit Knowledge)
- Main function implementation and application (Explicit Knowledge and Tacit Knowledge)

Each department at the department level has knowledge particular for this department based on the functions of the department, such as the experiences of the staff in the Product Management Department. The first and second stages are similar to the Enterprise level. There is one obvious problem in the department level, even in the group level. The problem is the knowledge in meeting minutes. This knowledge contains both explicit and some information about possible tacit knowledge. In the Enterprise level, meetings are important and intensive. Therefore, the meeting minutes for every meeting are recorded. However, in the department level, some meetings just involve a small number of people, and minutes are normally not recorded. It leads to a loss of knowledge. The knowledge management system should ensure all the knowledge in the meetings is managed in the system.

The typical example of explicit knowledge of the Product Management Department is the explanation of the department functions. The policies and rules are also the explicit knowledge of the department. The head of department and rest of staff need to follow the explanation and principles. The main function of the Product Management Department is to plan and design the production line.

The example of tacit knowledge includes the experiences and expertise of the department leader. Another tacit knowledge is in the communication between departments. This tacit knowledge is most forgettable tacit knowledge in the department level and group level. The methods of the communication between departments are multiple and might include emails, telephone calls, meetings and face to face charts. All this knowledge needs to be managed in knowledge management system.

In the group level, the knowledge analysis at group level is similar to the knowledge at department level, if the groups belong to the parent department, such as the Product Concept Development Group belonging to the Product Management Department. The small difference is that the knowledge in the group level is more specific to detailed tasks. The Product Concept Development Group of the Product Management Department is explained as an example. It can also show which kinds of knowledge is similar comparing with its parent department and which kind of knowledge is different.

The main output knowledge of the Product Concept Development group is the product concept report. This report is a typical generated knowledge in the group. The concept report contains the full description of the product concept including figures, models, processes and descriptions. It is the only way to output the product concept in the collaborating company. The knowledge involved in the group level is not only the input knowledge, but also the main part of the generated knowledge. Therefore, the requirements of managing the generated knowledge are necessary. The knowledge of the product concept group is shown as following:

- Generated product concept report (Explicit Knowledge)
- Relevant previous product concept report (Explicit Knowledge)
- Key group policies and rules (Explicit Knowledge)
- Relevant supporting process and models (Explicit and Tacit Knowledge)
- Experiences of staff of this level (Tacit Knowledge)
- Expertises of staff (Tacit Knowledge)
- Lessons learned (Tacit Knowledge)
- Supporting in-house product knowledge (Explicit and Tacit Knowledge)
- Communication records between departments and groups vertically and horizontally (Explicit and Tacit Knowledge)
- Problems address and possible solutions (Explicit and Tacit Knowledge)
- Group level Meetings minutes and reports, video and audio (Explicit and Tacit Knowledge)
- Standardisations (Explicit Knowledge)

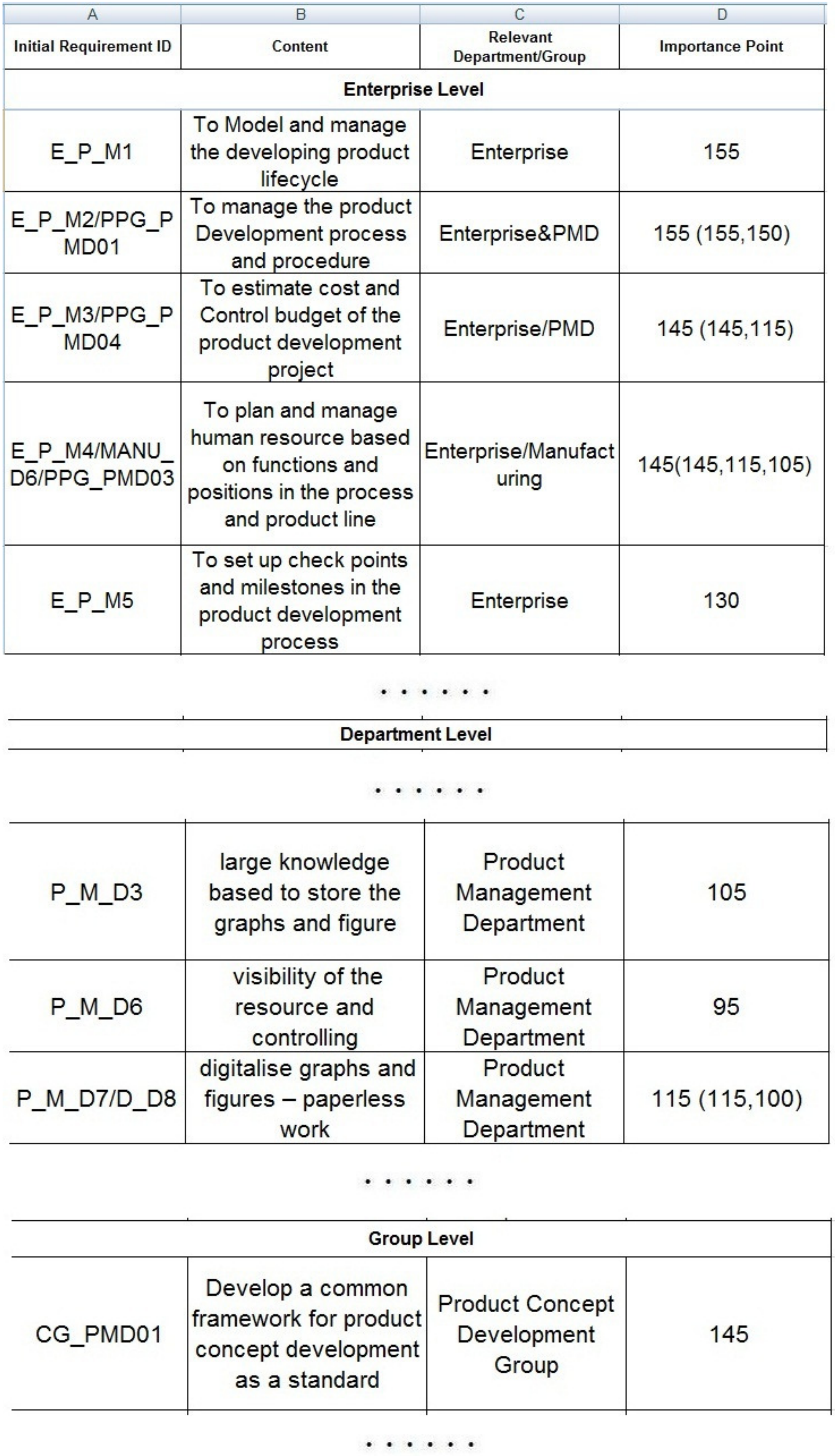
The clarified knowledge and users' requirements are then categorised. The method of categorising knowledge and users' requirements will be described and discussed in the next section.

6.3. Categorising Users' Requirements

The main aim of this section is to apply the "Categorise users' requirements and knowledge" task of the guideline in the collaborating company. In the analysing users' requirements stage, the users' requirements are classified based on the departments or groups originally. In this task, firstly the boundary of the departments and groups is broken down. All the requirements are assembled together in a uniform list as shown in Figure 6.4. To avoid some requirements being lost during the process, the list uses the Initial Requirement ID to ensure all user's requirements are included.

Figure 6.4 only shows part of the whole list due to the large size of the complete list. The whole list is provided in Appendix D. As discussed before, the overlaps and duplications are serious when the boundary of departments and group is broken. In this project, the overlaps and duplications not only exist in one single organisational level, but also exist between levels. For example, in Figure 6.4, the second requirement of the enterprise level (E_P_M2/PPG_PMD01), "To manage the product development process and procedure", is also required by the Product Planning Group of the Product Management Department. Therefore, these two requirements are combined as one requirement. There is a rule for merging the requirements. If there is a requirement at lower level which is also required by a higher level, the requirement will be combined and kept at higher level. Higher levels normally have more functions. However, if the needed functions can not satisfy the lower level, the duplicated requirements are still combined in the higher level with functional supplements from the lower level.

There are other situations in the overlaps and duplications. These situations are the overlaps or duplication just stay in one single level, such as the third requirement in the Department Level (P_M_D7/D_D8) in Figure 6.4. There are more than one department require "digitalise graphs and figures – paperless work". Therefore, this requirement can be combined in the Department Level. In this stage, all the requirements still contain the IDs as in the list of initial requirements. The combined requirements should keep all the relevant IDs of the initial requirements, in order to trace requirements from start to the end. In Figure 6.4, the Importance Point column contains all the points for each requirement.



A	B	C	D
Initial Requirement ID	Content	Relevant Department/Group	Importance Point
Enterprise Level			
E_P_M1	To Model and manage the developing product lifecycle	Enterprise	155
E_P_M2/PPG_PMD01	To manage the product Development process and procedure	Enterprise&PMD	155 (155,150)
E_P_M3/PPG_PMD04	To estimate cost and Control budget of the product development project	Enterprise/PMD	145 (145,115)
E_P_M4/MANU_D6/PPG_PMD03	To plan and manage human resource based on functions and positions in the process and product line	Enterprise/Manufacturing	145(145,115,105)
E_P_M5	To set up check points and milestones in the product development process	Enterprise	130

• • • • • •

Department Level			

• • • • • •

P_M_D3	large knowledge based to store the graphs and figure	Product Management Department	105
P_M_D6	visibility of the resource and controlling	Product Management Department	95
P_M_D7/D_D8	digitalise graphs and figures – paperless work	Product Management Department	115 (115,100)

• • • • • •

Group Level			
CG_PMD01	Develop a common framework for product concept development as a standard	Product Concept Development Group	145

• • • • • •

Figure 6.4 – Example of Categorised User's Requirements

As discussed in the last chapter, the new requirement which combined duplications should adopt the highest importance point of all the relevant requirements. For example, in Figure 6.4, the second requirement (E_P_M2/PPG_PMD1) of the Enterprise Level contains two initially captured user's requirements. One of the requirements gets the point of "150". The other initial requirement gets point of "155". Therefore, the highest point "155" is adopted as the importance point of this new requirement. It also keeps the original importance point of the initially captured requirements in the bracket, in order to show the number of involved knowledge users.

6.4. Categorising Captured Knowledge

As mentioned before, the collaborating company can be divided into 3 levels: enterprise level, department level and group level. Figure 6.5 shows the knowledge classification based on the 3 levels organisational structure in the collaborating company. There are 3 main kinds of knowledge existing in the company. Firstly, as shown at the left of Figure 6.5, there is knowledge shared by all levels. This kind of knowledge normally is the processes, models, standards and some other common knowledge in the company. The second category is that each level has its own specific knowledge. This kind of knowledge is not shared between levels for some reasons such as confidentiality. The third category is knowledge existing in the higher level and is shared by lower levels.

Therefore, the examples of the first category of the knowledge includes organisational principles and rules, standardisations, records of suppliers, contractors and partners, customer feedback and investigation of both existing and potential market. This category of knowledge is used and shared widely in 3 levels. For example, the records of suppliers are used in the enterprise level to help product manager decides suppliers which are adapted in the product development based on the records including prices, performance, efficiency and reliability. The records also can help departments to manage logistics and supply chain when the product line is designed and developed in the department level. In the group level, the manufacturing planning group needs the records to plan and control the manufacturing process and inventory. Therefore, to manage this category of knowledge the vertical links between levels have to be considered.

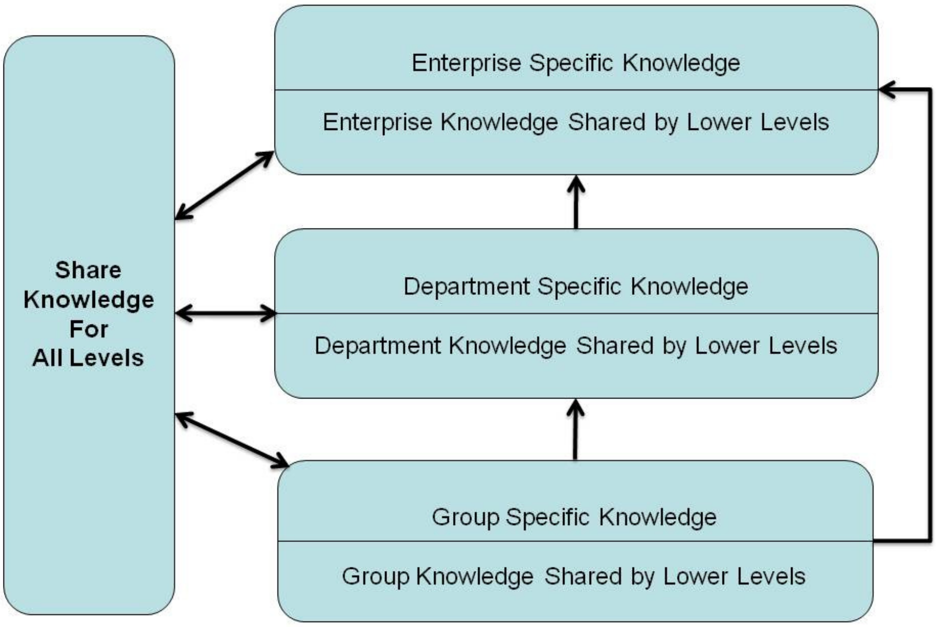

Share Knowledge For All Levels
Enterprise Specific Knowledge
Enterprise Knowledge Shared by Lower Levels
Department Specific Knowledge
Department Knowledge Shared by Lower Levels
Group Specific Knowledge
Group Knowledge Shared by Lower Levels

Figure 6.5 – Knowledge Categorisation

The second category of knowledge needs to be discussed and illustrated based on each level. In the enterprise level, the second category of knowledge includes experiences of product managers, board decisions which are based on the existing in-house product knowledge, in-house product reviews and achievements, some specific knowledge about what this product can bring and why, what potential benefit for a new product: technique and marketing, policy analysis and results. This category of knowledge is mainly used in the enterprise level. In the department level, as illustrated before, the Product Management Department includes experiences of staff including head of department, communication records between departments, key department functions' explanations and key principle and rules and main function implementation and application. In the group level, as illustrated before, the Product Concept Development Group, the specific knowledge is the generated product concept report which is the main function of this group. Therefore, the specific knowledge of each group is the main functions of the groups.

The third category of knowledge is based on the logical links between levels, such as the parent-child relationships between levels. The typical knowledge of the third category includes all the processes and product models. Processes and models can be extended and separated based on levels of the organisation and functions of each level. The requirement of processes and models is different at different levels. The enterprise level focuses on the whole process control and management. The details of processes and models normally not appeared in the enterprise level. Processes and models are broken down into details in the lower levels as sub processes and sub models. The departments and groups mainly focused on their own extended sub processes and sub models. The sub processes and sub models are actually linked in the same level. Therefore, the knowledge about processes and models need to be managed vertically and horizontally.

For all the 3 categorises, there are some explicit and tacit knowledge including: expertise experiences, lessons learned, meeting minutes, reports, video and audio, problems address and their possible solutions. All these knowledge should not be categorised based on its own right. For example, lessons learned are a typical example in the collaborating company. Any role in product development process has lessons learned in his/her daily work. The lessons learned can appear in any level in the knowledge system. This type of knowledge needs to be managed by each role which generates and uses the knowledge. Therefore, all listed knowledge should be managed based on each role who generates these kinds of knowledge. The detailed examples of the 3 categorises knowledge are provided in Appendix E.

There is another particular knowledge that needs to be considered. This knowledge is the communication records between departments and groups vertically and horizontally. In the enterprise level, product manager can communicate with the affiliated departments. This communication is one to many. In the department level, each department has to communicate with other departments horizontally. In the group level, each group not only has to communicate with the parent department vertically, but also communicates with other groups horizontally. This situation leads to complex knowledge flow and knowledge integration. Therefore, when the knowledge management system is designed and developed, the interacting and communication relationships have to be considered. This is a typical enhanced requirement which users did not mention in the users' requirements. There are some other enhanced requirements involved in, such as lessons learned, potential problems and solutions. All these enhanced requirements based on knowledge will be discussed in the next section.

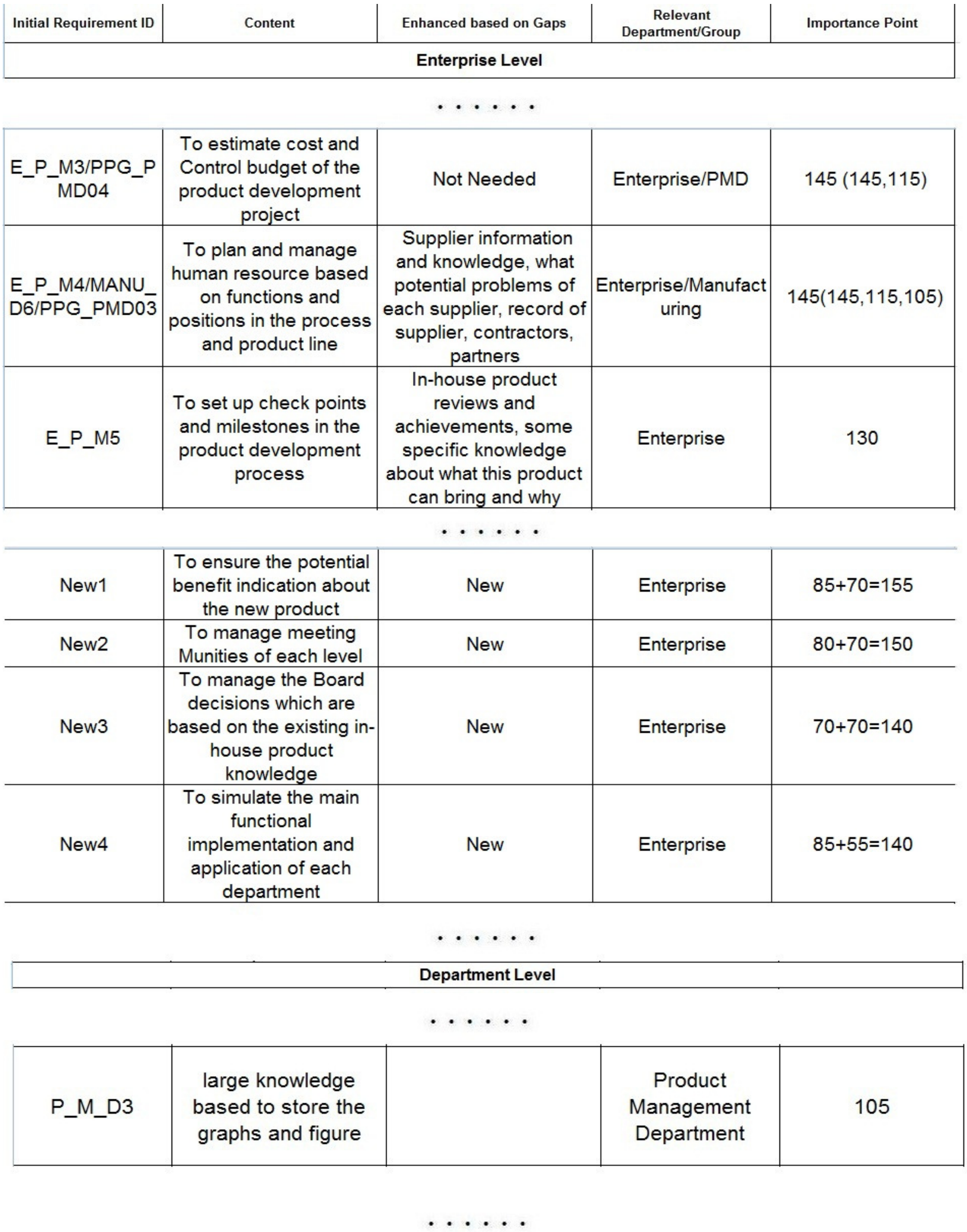
6.5. Integrate Captured Knowledge and Users' Requirements

After the "categorise users' requirements and knowledge" task, the knowledge needs to be integrated into users' requirements, in order to enhance the list of users' requirements, so that all knowledge can be managed including those indirectly specified in the users' requirements. In this section, the Product Management Department is explained as an example.

The easiest way to integrate knowledge into users' requirements is to compare knowledge and users' requirements, in order to find gaps in users' requirements. There are 3 kinds of situations in the comparison. The first situation is that the knowledge to be managed has already been included in the users' requirements. For example, one main function of the Product Management Department is to plan the product development process. Therefore, one of the basic requirements is to model the process in the system. The process and its sub-processes are the typical knowledge in the current industrial environment. In this situation the requirements in the list do not need to be changed.

The second situation is that there is not a requirement relevant to the knowledge at all in the users' requirements. The typical example is the meeting minutes. In the Product Management Department, nobody thinks that the meeting minutes are as important as knowledge. Management of this kind of knowledge should be added to the list of requirements.

The third situation is that certain requirements partly cover certain knowledge. In other words, there are more knowledge to be managed than the relevant users' requirements in the list. This situation is the most popular situation in the comparison. For example, one of the most popular requirements is the basic knowledge management requirements, such as knowledge storage and invoking. However, this basic requirement does not specify what types of knowledge. As previously discussed, managing explicit knowledge and tacit knowledge is not be the same. Therefore, a lot of details of knowledge should be specified in the requirements, such as, lessons learned, expertise experiences and so on. In this situation, the requirements in the list should be enhanced with more details about the knowledge to be managed.



Initial Requirement ID	Content	Enhanced based on Gaps	Relevant Department/Group	Importance Point
Enterprise Level				
• • • • • •				
E_P_M3/PPG_PMD04	To estimate cost and Control budget of the product development project	Not Needed	Enterprise/PMD	145 (145,115)
E_P_M4/MANU_D6/PPG_PMD03	To plan and manage human resource based on functions and positions in the process and product line	Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners	Enterprise/Manufacturing	145(145,115,105)
E_P_M5	To set up check points and milestones in the product development process	In-house product reviews and achievements, some specific knowledge about what this product can bring and why	Enterprise	130
• • • • • •				
New1	To ensure the potential benefit indication about the new product	New	Enterprise	85+70=155
New2	To manage meeting Munities of each level	New	Enterprise	80+70=150
New3	To manage the Board decisions which are based on the existing in-house product knowledge	New	Enterprise	70+70=140
New4	To simulate the main functional implementation and application of each department	New	Enterprise	85+55=140
• • • • • •				
Department Level				
• • • • • •				
P_M_D3	large knowledge based to store the graphs and figure		Product Management Department	105
• • • • • •				

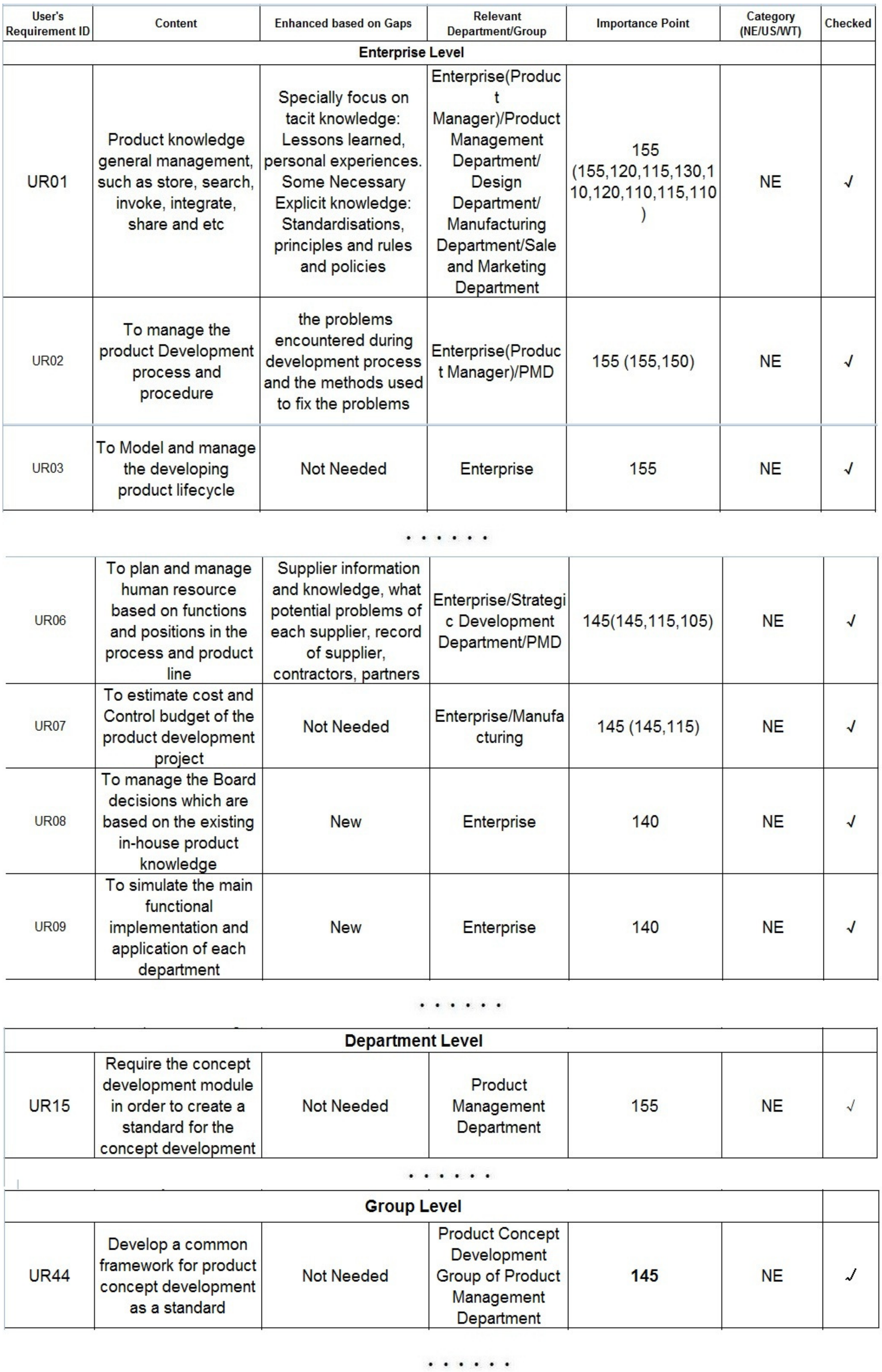
Figure 6.6 – Examples of Integrated Users' Requirements

Figure 6.6 shows part of the enhanced list of users' requirements. The third situation discussed above is that captured knowledge can be used to improve the statement of users' requirements, such as the first 3 rows in Figure 6.6. The third column is the improvement of the requirements. The second situation is the knowledge becomes a new requirement in the list, such as requirements with ID New1 to New4 in the list. These new requirements were initially ignored by knowledge users previously, and then they are generated based on the knowledge to be managed. The new requirements still need to be categorised and given importance point is the same was as other requirements. The integrated users' requirements will then be prioritised which will be discussed in the next section.

6.6. Prioritising Users' Requirements

After the integration of requirements and knowledge, all requirements are divided into 3 categories: necessary requirement, useful requirement and wanted requirements, in order to use the resource of the system maximally. The necessary requirement is the basic requirements which must be contained in the knowledge management system. The useful requirements need to be contained in the system to enhance the functions of the system, but the useful requirements are not compulsory in the design development. The wanted requirements are desired to be satisfied in the development of KM system. The necessary requirements must be satisfied when the system is developed. The useful requirements is better satisfied in the development of the system. However, if there are some difficulties such as large cost or resource waste, the useful requirements can be given up. The wanted requirements are optional to be satisfied based on the real situation of the system development.

The criterion of dividing requirements and the boundaries of the 3 categories are discussed and identified with both knowledge users and system developers in the collaborating company. As the last column in Figure 6.7, the point is in descending order. The importance point greater or equal to 120 is the necessary requirement (NE). The point between 100 and 120 inclusive is the useful requirement (US). The point below 100 is the wanted requirement (WT).



User's Requirement ID	Content	Enhanced based on Gaps	Relevant Department/Group	Importance Point	Category (NE/US/WT)	Checked
Enterprise Level						
UR01	Product knowledge general management, such as store, search, invoke, integrate, share and etc	Specially focus on tacit knowledge: Lessons learned, personal experiences. Some Necessary Explicit knowledge: Standardisations, principles and rules and policies	Enterprise(Product Manager)/Product Management Department/ Design Department/ Manufacturing Department/Sale and Marketing Department	155 (155,120,115,130,110,120,110,115,110)	NE	√
UR02	To manage the product Development process and procedure	the problems encountered during development process and the methods used to fix the problems	Enterprise(Product Manager)/PMD	155 (155,150)	NE	√
UR03	To Model and manage the developing product lifecycle	Not Needed	Enterprise	155	NE	√

• • • • • •

UR06	To plan and manage human resource based on functions and positions in the process and product line	Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners	Enterprise/Strategic Development Department/PMD	145(145,115,105)	NE	√
UR07	To estimate cost and Control budget of the product development project	Not Needed	Enterprise/Manufacturing	145 (145,115)	NE	√
UR08	To manage the Board decisions which are based on the existing in-house product knowledge	New	Enterprise	140	NE	√
UR09	To simulate the main functional implementation and application of each department	New	Enterprise	140	NE	√

• • • • • •

Department Level						
UR15	Require the concept development module in order to create a standard for the concept development	Not Needed	Product Management Department	155	NE	√

• • • • • •

Group Level						
UR44	Develop a common framework for product concept development as a standard	Not Needed	Product Concept Development Group of Product Management Department	**145**	NE	√

• • • • • •

Figure 6.7 – Example of the List of Users' Requirements

Figure 6.7 presents part of the list of user's requirements after integrated with captured knowledge. One of the main aims of the "Prioritised integrated Requirements" task is to rank all the requirements from high point to low based on each level. The users' requirements IDs are sequential. Therefore, the top requirements are certainly necessary requirements. The requirements in the middle are useful requirements. The bottom requirements are wanted requirements. The new knowledge users' requirements IDs are generated. The initial user's requirement IDs are also kept in the list, because this list of users' requirements needs to be rechecked and confirmed with the knowledge users. The list of users' requirements is processed through merging and restructuring by system developers. The representation of requirements is quite different comparing with the initial list of users' requirements. Therefore, keeping logical relationships between the initial users' requirements and the list of users' requirements is necessary for rechecking the list of users' requirements with knowledge users.

The output of this task is the list of requirements that is enhanced and prioritised. The "check user's requirements with users" task then checks and confirms the list of requirements with users. If there are any disagreements, misunderstandings and new additional requirements, that requirement will be reprocessed from task "Capture User's Requirements and knowledge" to task "Prioritise User's Requirements" until all the requirements are agreed by knowledge users and business leaders. The last column of Figure 6.7 represents the completion of the task. The final users' requirements will be used to develop system specification which will be discussed in the next chapter.

6.7. Summary

According to the research methodology in this project, there is a case study to verify and improve the developed methodology in the collaborating company. The whole case study is described in several chapters. This chapter aims to give an explanation for capturing and analysing the knowledge and users' requirements based on the guideline process. Many examples and real data are provided in this chapter. This chapter covers the process up to the final list of users' requirements in the guideline. In the next chapter, the list of requirements will be used to develop system functional requirements, and finally transferred to system specifications and implemented functions of system specifications.

Chapter 7: Developing Knowledge Management System Specifications with the Developed Methodology

In the last chapter, a case study of capturing and integrating knowledge and requirements was described. The output is a final list of users' requirements. In this chapter, the final list of users' requirements will be used to develop the KM system specification including following the guideline (as shown in Figure 7.1). The Product Management Department of the collaborating company is used as the case study.

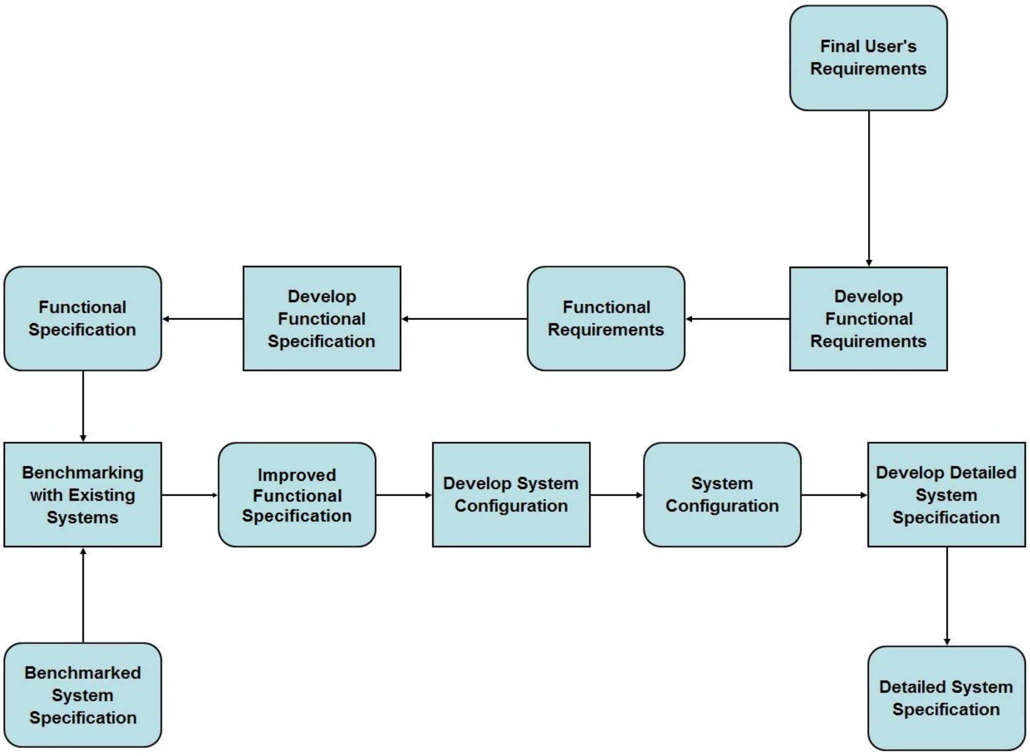
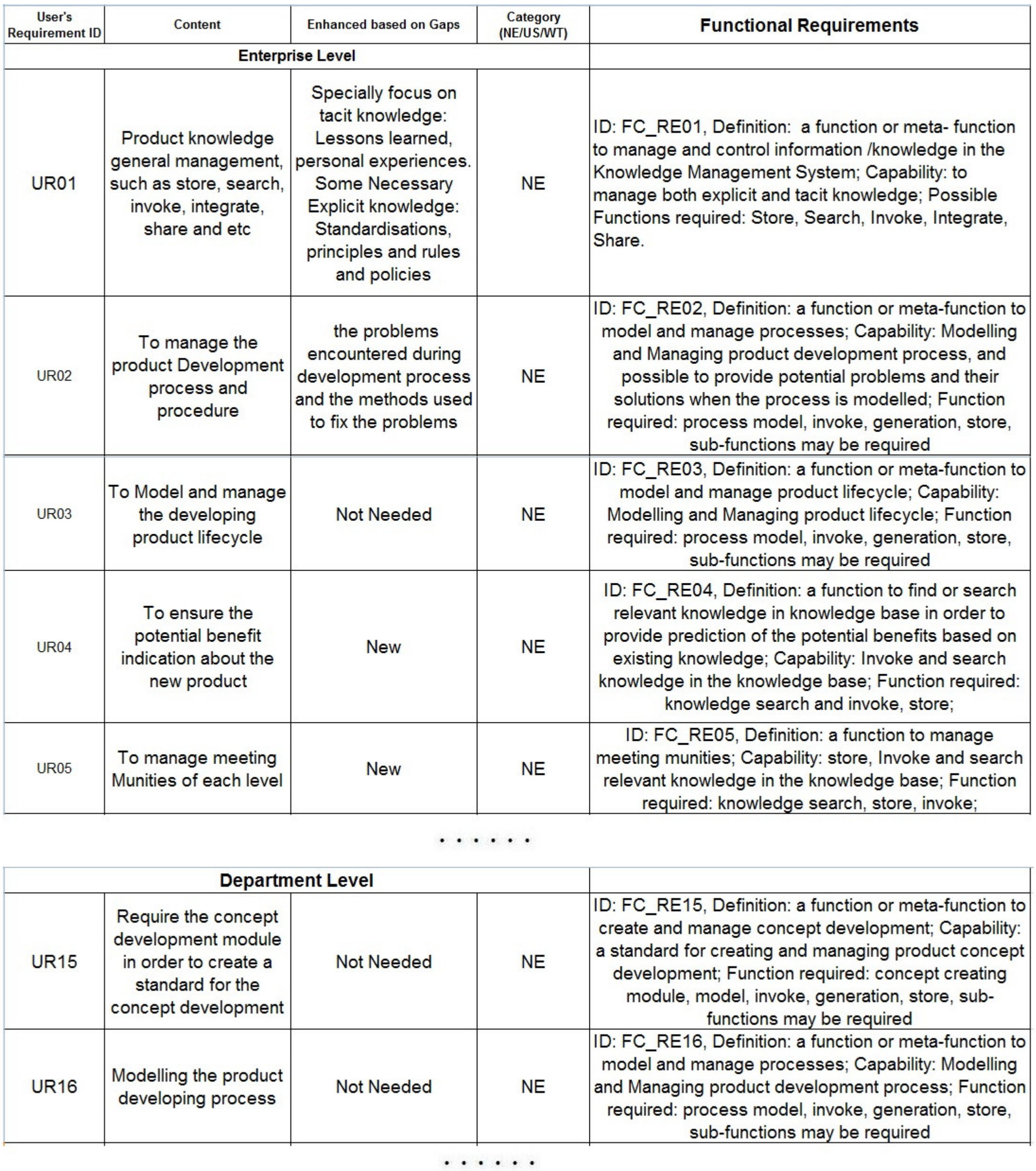

Final User's Requirements
Develop Functional Requirements
Functional Requirements
Develop Functional Specification
Functional Specification
Benchmarking with Existing Systems
Benchmarked System Specification
Improved Functional Specification
Develop System Configuration
System Configuration
Develop Detailed System Specification
Detailed System Specification

Figure 7.1 – The Guideline of Developing System Specification (Part of Figure 5.5)

7.1. Converting Users’ Requirements to Functional Requirements

According to the guideline of the developed methodology, the users’ requirements are converted to functional requirements. This conversion is one user’s requirement to one functional requirement. Functional requirements describe and define the possible functions and its effects in a system. Figure 7.2 show examples of converting users’ requirements to functional requirements.



User's Requirement ID	Content	Enhanced based on Gaps	Category (NE/US/WT)	Functional Requirements
Enterprise Level				
UR01	Product knowledge general management, such as store, search, invoke, integrate, share and etc	Specially focus on tacit knowledge: Lessons learned, personal experiences. Some Necessary Explicit knowledge: Standardisations, principles and rules and policies	NE	ID: FC_RE01, Definition: a function or meta- function to manage and control information /knowledge in the Knowledge Management System; Capability: to manage both explicit and tacit knowledge; Possible Functions required: Store, Search, Invoke, Integrate, Share.
UR02	To manage the product Development process and procedure	the problems encountered during development process and the methods used to fix the problems	NE	ID: FC_RE02, Definition: a function or meta-function to model and manage processes; Capability: Modelling and Managing product development process, and possible to provide potential problems and their solutions when the process is modelled; Function required: process model, invoke, generation, store, sub-functions may be required
UR03	To Model and manage the developing product lifecycle	Not Needed	NE	ID: FC_RE03, Definition: a function or meta-function to model and manage product lifecycle; Capability: Modelling and Managing product lifecycle; Function required: process model, invoke, generation, store, sub-functions may be required
UR04	To ensure the potential benefit indication about the new product	New	NE	ID: FC_RE04, Definition: a function to find or search relevant knowledge in knowledge base in order to provide prediction of the potential benefits based on existing knowledge; Capability: Invoke and search knowledge in the knowledge base; Function required: knowledge search and invoke, store;
UR05	To manage meeting Munities of each level	New	NE	ID: FC_RE05, Definition: a function to manage meeting munities; Capability: store, Invoke and search relevant knowledge in the knowledge base; Function required: knowledge search, store, invoke;

· · · · · ·

Department Level				
UR15	Require the concept development module in order to create a standard for the concept development	Not Needed	NE	ID: FC_RE15, Definition: a function or meta-function to create and manage concept development; Capability: a standard for creating and managing product concept development; Function required: concept creating module, model, invoke, generation, store, sub-functions may be required
UR16	Modelling the product developing process	Not Needed	NE	ID: FC_RE16, Definition: a function or meta-function to model and manage processes; Capability: Modelling and Managing product development process; Function required: process model, invoke, generation, store, sub-functions may be required

· · · · · ·

Figure 7.2 – Example of Conversion from Users’ Requirements to Functional Requirements

There are two steps of converting users' requirements to functional specifications. Firstly, each user's requirement is converted to a functional requirement. Then, the converted functional requirements improved to be functional specification through removing all duplications and overlaps between functional requirements. In Figure 7.2, the users' requirements ID and contents are also kept, in order to keep links in transferring. The right column shows converted functional requirements. Unfortunately, there is not a standard structure of the system specification in the collaborating company. System developers create a template for better developing functional requirements and further continuous development of the knowledge management system. The functional requirement template identifies several aspects, such as functional requirement ID, definition of functional requirements, capability of the functions and possible functions required in the system design.

When the users' requirements are converted to the functional requirements, there are some obvious duplications and overlaps between functional requirements. For example, in Figure 7.2, the second functional requirement (FC_RE02) and the third functional requirement (FC_RE03) are both required to model and manage process. FC_RE02 requires to manage development process, and FC_RE03 requires to model and manage product lifecycle. They are actually same requirement in the system development. One meta function can satisfy all requirements for modelling and managing process.

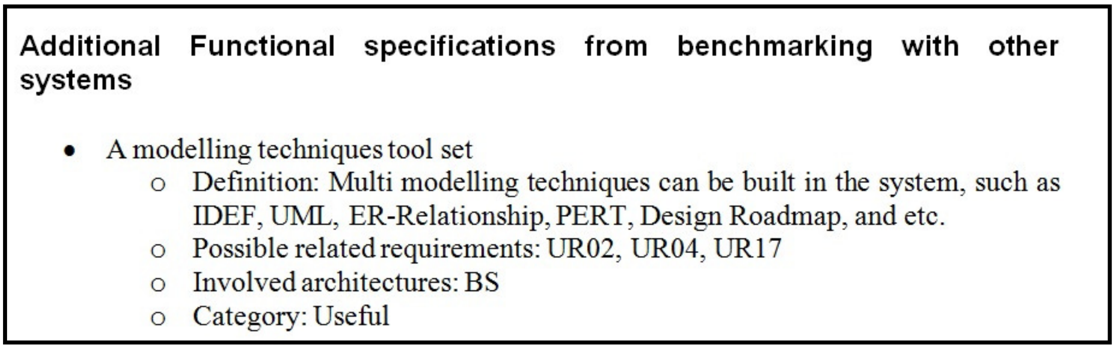
Another example is the functional requirement (FC_RE05) about to manage meeting minutes. It actually is required to manage the knowledge in the meeting of each level of the enterprise. From the system development point of view, the meeting minutes is a kind of knowledge representation. Managing meeting knowledge is actually managing a kind of knowledge in the system. Therefore, in the development of KM system, FC_RE05 can be combined with FC_RE01which is the generally required to manage all knowledge in the collaborating company.

According to the guideline (as shown in Figure 7.1), the functional specification (full version of functional specification will be provided in Appendix F) is used to benchmark with existing systems, such as PLM system, in order improve and enhance functional specifications to improved functional specifications which will be discussed in next sections.

7.2. Benchmarking the Developed System Specification

According to the guideline (as shown in Figure 7.1), the developed functional specifications are benchmarked against the existing systems in the current market. The relevant knowledge of the existing system can be found in the open-source system development, or it can be bought from other suppliers. The typical example of this is TOGAF. The Open Group provides a framework for companies which wish to build their own systems. The Open Group also provides opportunities to show the successful cases for ongoing system developers.

Figure 7.3 shows an example of the improved functional specification from the benchmarking excise with additional (useful) functions from other existing systems. The example describes a function which is included in a developed system based on TOGAF. In that system, most of modelling techniques are included in, such as UML (Unified Modelling Language), ER (Entity Relationship), Design Roadmap and IDEF (ICAM Definition Languages). Users can use these modelling techniques without using other external modelling tools. In other words, users can finish modelling within the system. This function can save extra costs for purchasing software and their licences in the company. Therefore, it is a very useful function which can be included in the knowledge management system in the collaborating company.



Additional Functional specifications from benchmarking with other systems

- A modelling techniques tool set
    - Definition: Multi modelling techniques can be built in the system, such as IDEF, UML, ER-Relationship, PERT, Design Roadmap, and etc.
    - Possible related requirements: UR02, UR04, UR17
    - Involved architectures: BS
    - Category: Useful

Figure 7.3 – An Example of Additional Functional Specification from Benchmarking

According to the guideline (as shown in Figure 7.1), after benchmarking against with existing systems, the improved functional specifications are formed. All these improved functional specifications need the confirmation of knowledge users. Knowledge users may not agree or have concern about the additional functional specifications from benchmarking. Therefore, there is a possibility to delete some improved functional specifications. After all the additional functional specifications are confirmed with knowledge users, the system configuration will be developed based on the improved functional specifications. There are more details about benchmarking will be provided in Appendix G.

7.3. Developing System Configuration

System configuration is normally a structured diagram to illustrate the functional structure of the system. In some companies, system configuration is included in the system specifications, because it provides a capability boundary for system development. In this project, system configuration provides a structured diagram as a perspective to all the roles involved in the system development.

Figure 7.4 shows the system configuration of the knowledge management system designed for the collaborating company. According to TOGAF, the whole system includes 3 main architectures: the Business Architecture, the Information System Architecture and the Technology Architecture. The Information architecture can be divided into 2 other architectures: the Application Architecture and Data Architecture. In the system configuration structure diagram, the whole system is set up based on these 4 Architectures.

As shown in Figure 7.4, the members of the collaborating (Users) can manage the business and daily work through the in the Business Architecture. The users are part of the Business Architecture, because the Business Architecture considers the business processes and the organisational structure. Users assess the system through interfaces which links to the Application Architecture. The business requirements are applied and implemented as applications in the Application Architecture. For example, there is a business requirement for managing the process through the system in the Product Management Department. The process application can satisfy the business requirement about managing process in the Product Management Department, such as process modelling, process optimising, hierarchical process managing and so on. All of these applications can be set up in the Application Architecture.

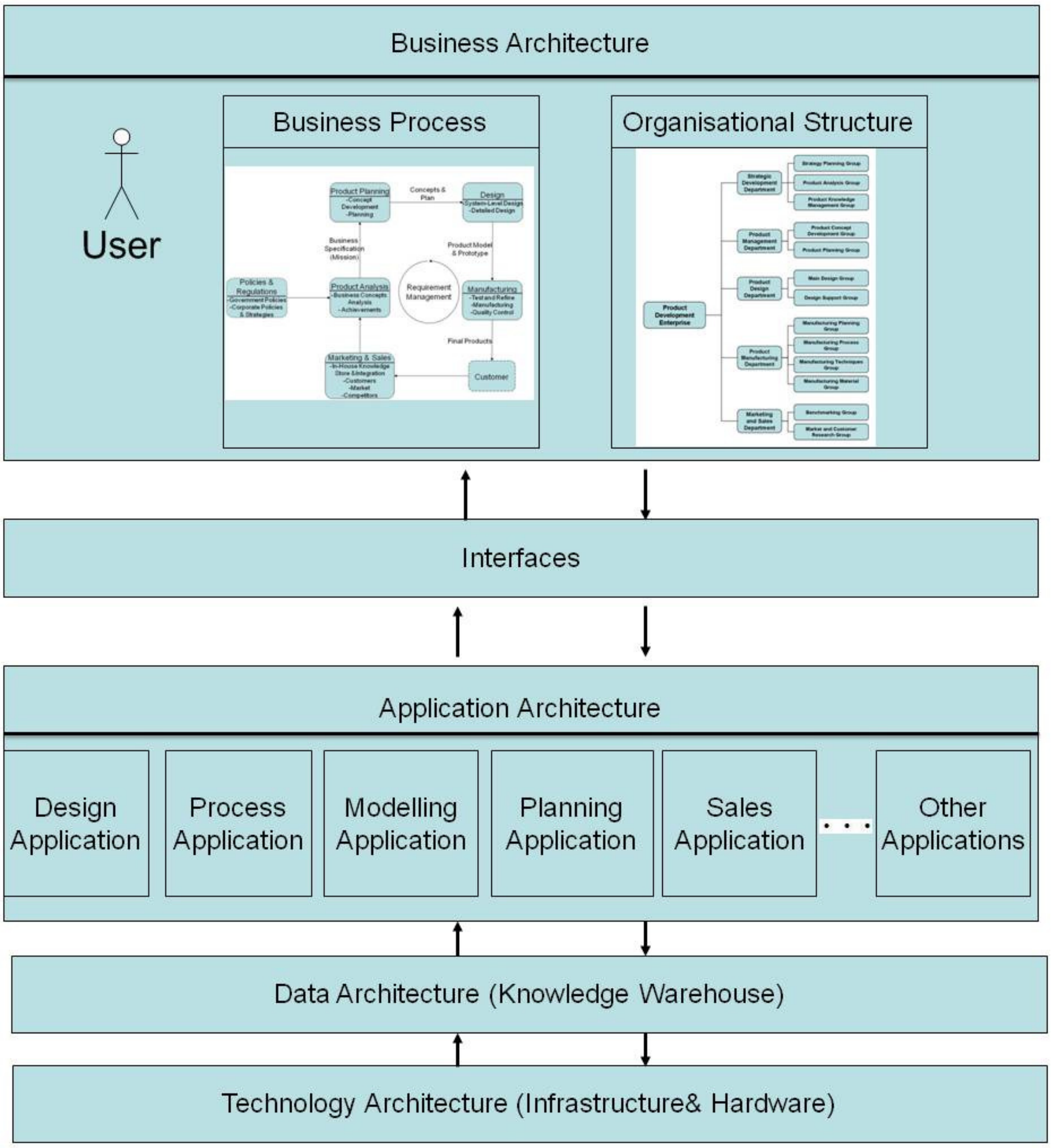


Figure 7.4 – System Configuration Designed for the Collaborating Company

The Application Architecture contains both interface and applications in the structure. In the Application Architecture, there are some other applications about the daily work of the Product Management Department rather than purely business requirements. The daily work is used to manage some tacit knowledge or some knowledge which is easy to forget in the daily work. For example, the lessons learned is a typical tacit knowledge in the daily work in the Product Management Department. Each role should have the lessons learned from daily work. The knowledge management system allows the users to summarise their lessons learned through the system application, in order to manage this tacit knowledge in the system.

The Application Architecture is linked to the Data Architecture, which provides an environment for managing knowledge from different applications. The Data Architecture can integrate all the relevant knowledge together, in order to better manage knowledge centrally, for sharing. The Data Architecture not only provides a space to store knowledge, but also provides a method for storing knowledge. For example, each type of knowledge has a particular sub knowledge warehouse to store and manage. The process knowledge is stored and managed in the process knowledge warehouse. The number of the sub knowledge warehouse is not fixed, and it can be changed depending on the situation in the enterprise.

At the bottom of the system configuration, it shows the Technology Architecture which is the Infrastructure and Hardware of the system. It provides infrastructure technology to the system. The purpose of the system configuration is to provide a certain perspective and standard for each role involved in system development. It can reduce misunderstanding and mistakes during the whole process of the system development. After developing system configuration, the final output which is the system specification will be developed based on the improved functional specifications and the system configuration.

7.4. Developing System Specification

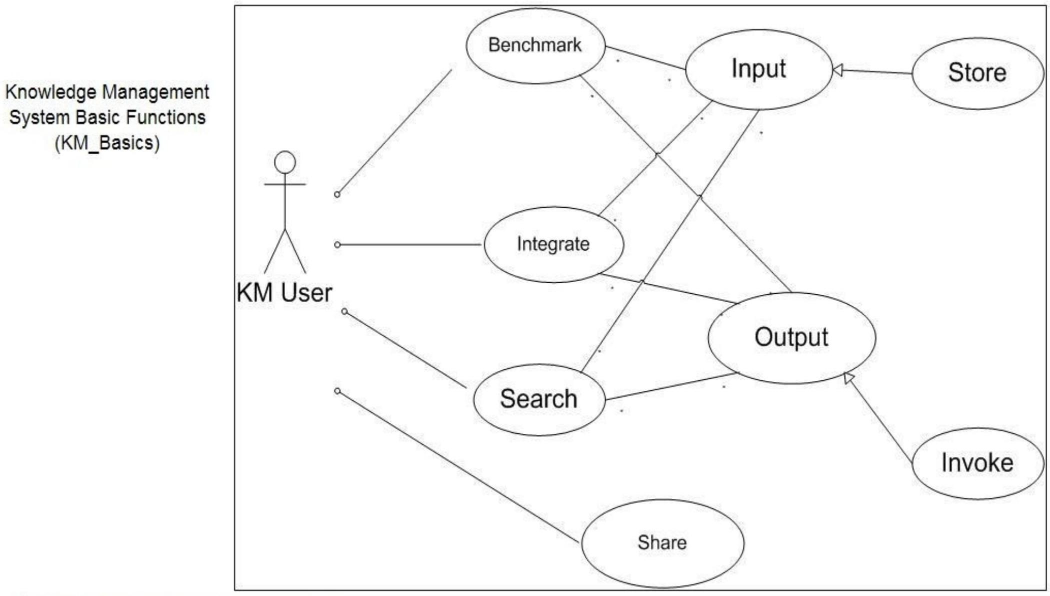
System specification is a main description of the information/KM system to be developed. Programmers can follow system specifications to develop the system. In this section, the procedure of developing system specification will use FC_RE01 and FC_RE02 (shown in Figure 7.2) as the illustrations.

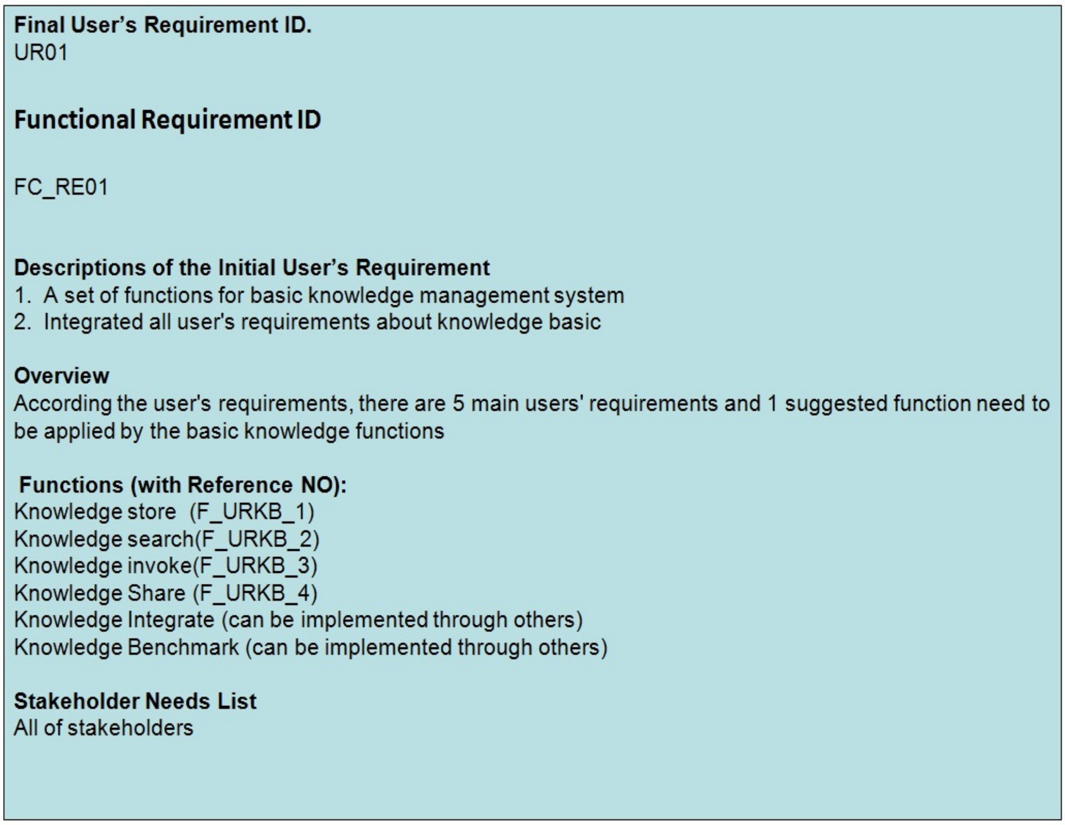
The procedure of developing system specification contains two steps. The first step is to develop the design details of the system. Some researchers also call this step is to develop system design specification (Levin, 2006). System design specification provides a specific description of system design and its process. In other words, system design specification is to extend system specification to details based on functional requirements. System design specification usually contains several features of the system, such as the purpose of the design, usability of the system, functional characteristics of the system, presentation of the system and resources of the system including budget and cost. System design specification visualises figures, tables, templates and features based on initial user's requirements, in order to help whole system development to be unambiguous. The second step is to develop functions of the system based on the design specification. These developed functions can be implemented directly by programmers. The second step will be described and discussed latterly in this section.

According to the functional requirements, the most important functional requirement is the basic function for managing knowledge in the knowledge management system including knowledge storage, knowledge integration, knowledge invoking, knowledge search and knowledge sharing (FC_RE01 in Figure 7.2). There is a suggested function, benchmarking, that can be implemented within this system design specification. The benchmarking function is implemented through invoking knowledge from the system and input new generated knowledge into the system. Therefore, it can be implemented in the system design specification.

Figure 7.5 presents the system design specification for basic knowledge management functions (KM_Basics for short). The system design specification contains the name, UML diagram and description. The name is at the top-left of Figure 7.5 which includes system design specification name and the system design specification ID in brackets. The UML diagram is in the top-right of Figure 7.5. The description is at the bottom of Figure 7.5.

In the top-right of Figure 7.5, the UML diagram represents the structure and relationships from the user-end to the basic functions of the system. In the basic knowledge management system design, all the knowledge user's requirements can be divided into two types: put some knowledge into the system, and take knowledge out of the knowledge system, i.e., “input” and “output” of a system. The “input” function includes “store” function which is input knowledge into the system. The “output” function includes “invoke” function which is system output knowledge to the users.




Final User’s Requirement ID.
UR01

Functional Requirement ID

FC_RE01

Descriptions of the Initial User’s Requirement
1. A set of functions for basic knowledge management system
2. Integrated all user's requirements about knowledge basic

Overview
According the user's requirements, there are 5 main users' requirements and 1 suggested function need to be applied by the basic knowledge functions

Functions (with Reference NO):
Knowledge store (F_URKB_1)
Knowledge search(F_URKB_2)
Knowledge invoke(F_URKB_3)
Knowledge Share (F_URKB_4)
Knowledge Integrate (can be implemented through others)
Knowledge Benchmark (can be implemented through others)

Stakeholder Needs List
All of stakeholders

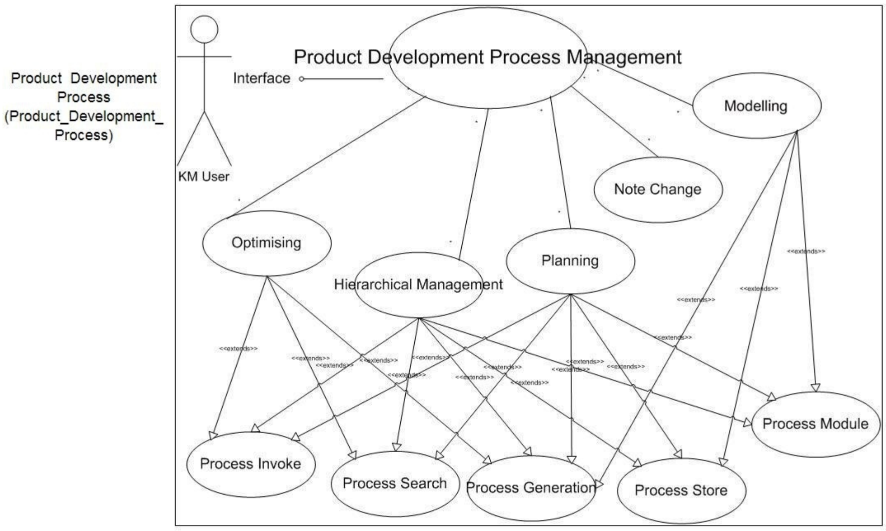
Figure 7.5 – Design Specification for Basic Knowledge Management Functions

Some functions are implemented indirectly, such as “knowledge integrate” and “benchmarking”. This type of functions is implemented through several basic functions. For example, the “knowledge integrate” function needs to input and output knowledge to/from the knowledge system. Based on the current technology and the circumstance of the collaborating company, the knowledge is integrated manually rather than automatically by the knowledge system. With the ideal assumption, knowledge user only stores the knowledge in the knowledge system database. Therefore, when knowledge is required to be integrated, the required knowledge needs to be found and invoked by the user first. The User integrates a set of knowledge to a new knowledge, and then the new knowledge is stored in the knowledge system to complete the process of knowledge integration.

There is another type of functional design which is purely system-level activities, such as the “knowledge share” function. The “knowledge share” function just needs system to create a function inside the system. In other words, there is not any activity from users.

The third type of function in the system design specification is the functions containing both user activity and system activity, such as the “knowledge search” function. Firstly, user needs knowledge. Therefore, user input a query for searching knowledge. This is an activity from user. Secondly, the system receives the query for searching knowledge. Therefore, the system searches the relevant knowledge instructed by the query. This is the system activity. Finally, the system displays a list of relevant knowledge to the user. The user can find the most appropriate knowledge in the list.

At the bottom of Figure 7.5, there is the description of the system design specification. The purpose of the description is to formalise the system design specification within a standard, in order to unify all system developers in the development process. The description contains 5 elements. The first element is “Final user’s requirement ID” to notice all the user’s requirements which are involved in this system design specification. The “Description of the initial user’s requirement” element is to analyse requirements and identify a method to design them. The “overview” element represents a brief description for the design specification, in order to clarify the basic design idea in the design specification. The “functions” element represents the possible functions that should be developed based on the design specification. These functions will be developed in the second step. The last content in the description is to list and identified stakeholders which are actually initial knowledge users involved in the system design specification.



Initial User's Requirement NO.
UR02,UR04

Descriptions of the Initial User's Requirement
-Integration of all requirements of the product process management
- the problems encountered during development process and the methods used to fix the problems
-Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners

Overview

According the list of user's requirements, there are several user's requirements about the product development process. The integrated user's requirements includes the functions which shown as the left figure

Functions (with Reference NO):
Process Modelling (F_UR_PROCESS1)
Low-level sub-functions
- Process Module
- Process Store
- Process Generation

Process Optimising(F_UR_PROCESS2)
Low-level sub-functions
- Process Search
- Process Store
- Process Invoke

Process Planning (F_UR_PROCESS3)
Low-level sub-functions
- Process Module
- Process Store
- Process Generation
- Process Search
- Process Invoke

Hierarchical Management (F_UR_PROCESS4)
Low-level sub-functions
- Process Module
- Process Store
- Process Generation
- Process Search
- Process Invoke

Note Change (F_UR_PROCESS5)

Stakeholder Needs List
Product manager, Product Management Department, Design Department, Manufacturing Department

Figure 7.6 – System Design Specification for the Product Development Process

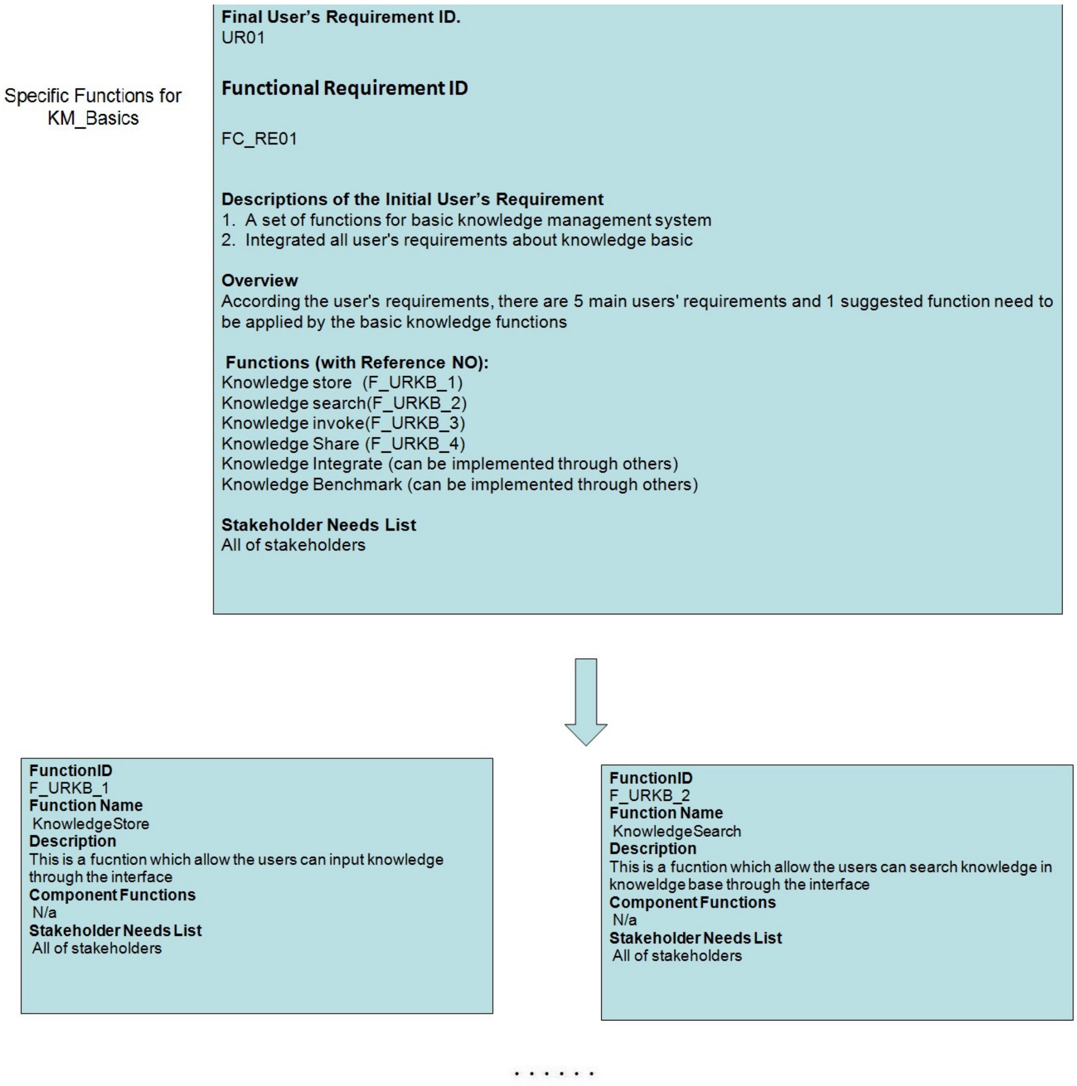
The basic knowledge system function is one kind of the system design specifications based on the basic requirements of the knowledge management system. There are some other requirements which focus on the main business in the collaborating company. In this project, new product development is focused and adopted as the main business with the defined enterprise. Therefore, the new product development process is significant in the collaborating company. Many users' requirements are related to product development process. Figure 7.6 presents the system design specification of the product development process.

The representation of the system design specification of the product process development ("Product_Development_Process") is the same as the representation of the "KM_Basics" system design specification. The diagram shows that the knowledge user can manage product development process through a set of interfaces. The main functions of the product development process contains process modelling, process optimisation, process generation, process hierarchical management and note change based on the user's requirements.

Comparing with the "KM_Basics", the main functions of the "Product_Development_Process" may contain several sub-functions. On the other hand, a sub-function can also be included in several different main functions. The purpose of this kind of structure is to save system resources. However, the inter-relationships are complex in the system as shown in Figure 7.6.

The arrows represent the main functions that can be extended to several sub-functions. For example, the product development process and its sub-processes need to be planned before launching in the real industrial environment. The "process planning" function mainly represents the links between knowledge users and the system. When a knowledge user wants to plan a process in the product development, firstly the user tries to find the relevant process in the knowledge system, then the relevant process will be found in the knowledge system. The "process search" sub-function and "process invoke" sub-function can satisfy the user seeking the relevant process. Secondly, the knowledge user can plan the new process in the system based on the "process module" sub-function which provides all the resources of the process techniques. The knowledge user can change anything with the "process module" sub-function. When the process is finished, it can be stored in the system to backup the ideas of the process temporarily with "process store" sub-function, or it can be generated with "Process generation" sub-function and then stored in the knowledge base. When the process is saved in the system, it can be shared with others who have authority to access it. The design theory of other main functions is the same as the “process planning” sub-function.

The description template of this system design specification is unified in the whole system design specification stage. Therefore, the structure of the “Product_Development_Process” is the same as the “KM_Basics”. The difference is that the “Function” part needs to clarify the sub-functions of each main function (as shown in the right-bottom of Figure 7.6).



Specific Functions for KM_Basics

Final User’s Requirement ID.
UR01

Functional Requirement ID

FC_RE01

Descriptions of the Initial User’s Requirement
1. A set of functions for basic knowledge management system
2. Integrated all user's requirements about knowledge basic

Overview
According the user's requirements, there are 5 main users' requirements and 1 suggested function need to be applied by the basic knowledge functions

Functions (with Reference NO):
Knowledge store (F_URKB_1)
Knowledge search(F_URKB_2)
Knowledge invoke(F_URKB_3)
Knowledge Share (F_URKB_4)
Knowledge Integrate (can be implemented through others)
Knowledge Benchmark (can be implemented through others)

Stakeholder Needs List
All of stakeholders

FunctionID
F_URKB_1
Function Name
KnowledgeStore
Description
This is a fucntion which allow the users can input knowledge through the interface
Component Functions
N/a
Stakeholder Needs List
All of stakeholders

FunctionID
F_URKB_2
Function Name
KnowledgeSearch
Description
This is a fucntion which allow the users can search knowledge in knoweldge base through the interface
Component Functions
N/a
Stakeholder Needs List
All of stakeholders

. . . . . .

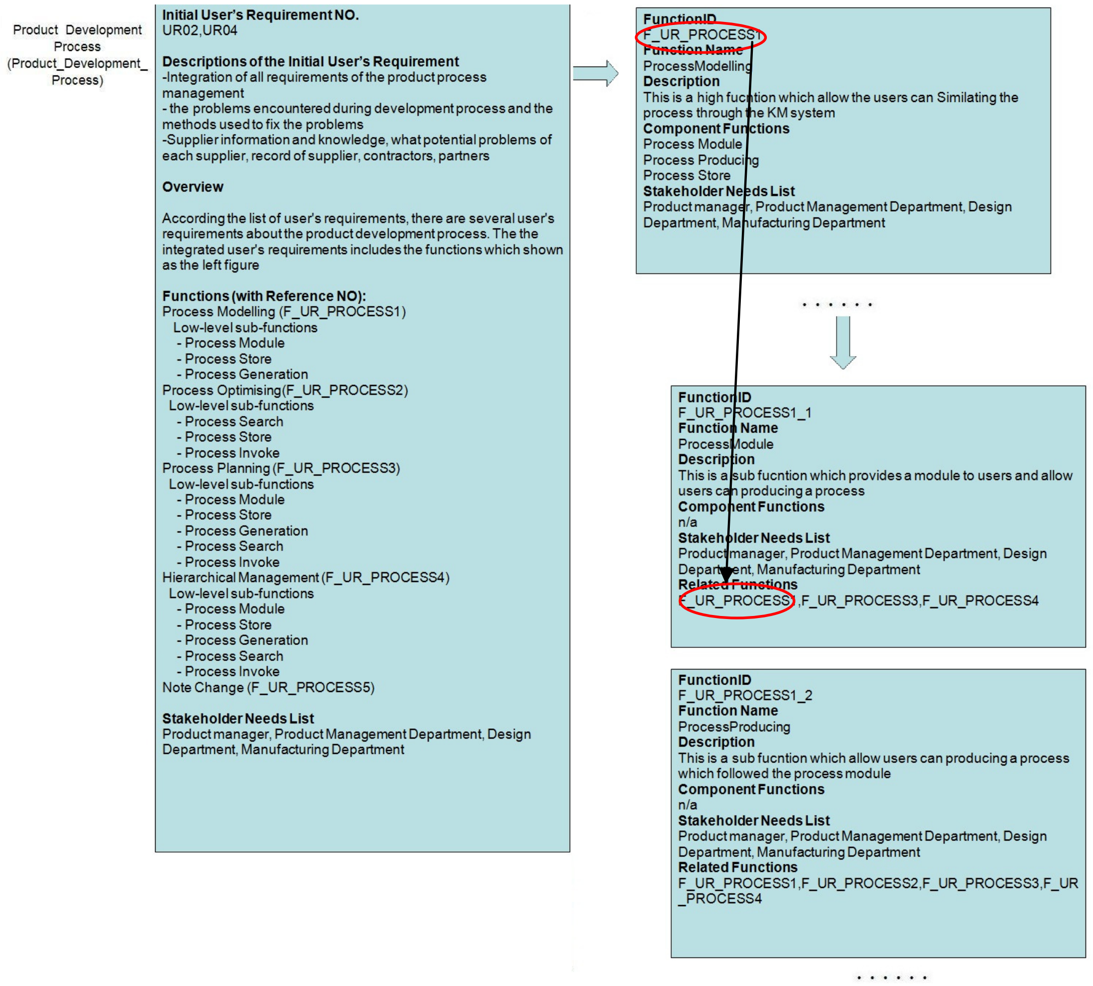
Figure 7.7 – Specific Functions of KM_Basics in System Specification

There are two examples used to describe how the system specification is generated. The second step is to develop specific functions of the system based on the system design specifications. Figure 7.7 presents part of the development of functions in the system specification about “KM_Basics”. It contains the name of the specific functions (shown at the top left), the system design specification (shown in the top right) and developed specific functions (shown at the bottom) in the diagram.

The system functions are developed based on the system design specification. In the system design specification of the KM_Basics, the design suggests 4 basic functions that should be developed as the systems functions. Each function has its own function ID to avoid overlaps and loss of functions. Another purpose of function ID is to clarity and trace requirements from users to system. The specific functions contain all the functions that are required to be developed. For example, the “knowledge store” functions with ID “F_URKM_1” is represented at the bottom left side (as shown in Figure 7.7). The attributes of the function include its ID, name, description, possible sub-function and relevant stakeholders. All of these attributes need to be considered when the function is implemented. The version of the diagram will be provided in Appendix H.

Figure 7.8 represents the transferring process of the “Product_Development_Process” which contains sub-functions. The structure is similar to the transferring process “KM_Basics”, because of the unified structure for every transferring process. The difference is that the transferring process of the “Product_Development_Process” includes one more column which represents the sub-functions.

Normally, the relationship between a function and its sub-functions is one to many. Therefore, all the sub-functions of each main function should be developed based on this relationship like a family tree to trace the relationship easily. However, the transferring process is not exactly following the system design specification in the “Product_Development_Process”. The diagram (Figure 7.6) of the system design specification describes complex relationships among main functions and their sub-functions. This complex relationship leads to an unclear structure for programmers and waste of the system resources due to the duplicates and overlaps of the sub-functions.



Product Development Process (Product_Development_Process)

Initial User's Requirement NO.
UR02,UR04

Descriptions of the Initial User's Requirement
-Integration of all requirements of the product process management
- the problems encountered during development process and the methods used to fix the problems
-Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners

Overview

According the list of user's requirements, there are several user's requirements about the product development process. The the integrated user's requirements includes the functions which shown as the left figure

Functions (with Reference NO):
Process Modelling (F_UR_PROCESS1)
Low-level sub-functions
- Process Module
- Process Store
- Process Generation
Process Optimising(F_UR_PROCESS2)
Low-level sub-functions
- Process Search
- Process Store
- Process Invoke
Process Planning (F_UR_PROCESS3)
Low-level sub-functions
- Process Module
- Process Store
- Process Generation
- Process Search
- Process Invoke
Hierarchical Management (F_UR_PROCESS4)
Low-level sub-functions
- Process Module
- Process Store
- Process Generation
- Process Search
- Process Invoke
Note Change (F_UR_PROCESS5)

Stakeholder Needs List
Product manager, Product Management Department, Design Department, Manufacturing Department

FunctionID
F_UR_PROCESS1
Function Name
ProcessModelling
Description
This is a high fucntion which allow the users can Similating the process through the KM system
Component Functions
Process Module
Process Producing
Process Store
Stakeholder Needs List
Product manager, Product Management Department, Design Department, Manufacturing Department

. . . . . .

FunctionID
F_UR_PROCESS1_1
Function Name
ProcessModule
Description
This is a sub fucntion which provides a module to users and allow users can producing a process
Component Functions
n/a
Stakeholder Needs List
Product manager, Product Management Department, Design Department, Manufacturing Department
Related Functions
F_UR_PROCESS1,F_UR_PROCESS3,F_UR_PROCESS4

FunctionID
F_UR_PROCESS1_2
Function Name
ProcessProducing
Description
This is a sub fucntion which allow users can producing a process which followed the process module
Component Functions
n/a
Stakeholder Needs List
Product manager, Product Management Department, Design Department, Manufacturing Department
Related Functions
F_UR_PROCESS1,F_UR_PROCESS2,F_UR_PROCESS3,F_UR_PROCESS4

. . . . . .

Figure 7.8 – Functions in System Specifications of Product_Development_Process

The principle of transferring main function to sub-functions is to avoid the duplicates and overlaps in sub-functions. According to the system design specification diagram in Figure 7.6, there are 5 sub-functions which are used by the main functions repeatedly in the system design specification. Therefore, the 5 sub-functions are extended. The relationships are not represented obviously in the main functions and sub-functions. There is an attribute named "related functions" in the sub-function (red ellipse in Figure 7.8). This attribute represents the relationships between the sub-function and main functions which uses this sub-function (red ellipses and arrow in Figure 7.8). This method not only avoids the duplicates and overlaps in the sub-functions, but also provides an understandable relationship between main functions and sub-functions.

7.5. Summary

In this chapter, the procedure of developing the system specification, which is the final output of applying the developed methodology, has been described and discussed. The guideline provides an opportunity for developing the system specifications based on the final users' requirements. The programmers can follow the system specifications to implement the knowledge management system.

The system specification will be evaluated and compared with some other existing systems in the current market in the next chapter.

Chapter 8: Evaluation of the Developed Methodology

In previous chapters, the developed methodology has been described with a comprehensive case study explaining how to use it. In this chapter, the methodology will be evaluated with the knowledge users in the collaborating company, and will be compared with the Product Lifecycle Management (PLM) system currently used in the company.

Firstly, the current IT solution in the collaborating company will be described and discussed. The advantages and limitations of the current IT solution in the company will be assessed. Then the developed methodology will be evaluated with knowledge users and IT support team in the collaborating company, in order to identify advantages and shortcomings of the developed methodology with respect to the current IT solution in the company. Finally, the benefits and achievements of the developed methodology will be pointed out.

8.1. The Current IT Solution in the Collaborating Company

During this project, the collaborating company implemented a Product Lifecycle Management (PLM) system which is from Siemens (Siemens, 2011) for its product development, in order to manage the product information centrally. The PLM system helps the collaborating company to share product information and cooperation between departments and groups.

Before the collaborating company implemented PLM system, the product development team still used personal computer based technically simple IT systems to store information and data. If some product model or design needs to be discussed between departments or groups, it has to be printed. This situation leads to 3 main limitations. The first one is that there are many mistakes based on the paper printing and human delivery. A lot of information can not be delivered in the right position at the right time. It leads to many unnecessary revised works during this procedure. The second limitation is that many leaders in different enterprise levels cannot obtain the required information directly. It leads to misunderstanding between the board, departments and groups. The third limitation is that the information is kept individually. A lot of information is overlapped and duplicated. Some information is lost. It leads to extra cost and unnecessary waste of resources. All these limitations lead to the low effectiveness and low efficiency during product development.

The PLM system can overcome the 3 limitations mentioned above. The PLM system provides an opportunity to ensure that the required information can be delivered correctly and in time. Users can access the required information through the PLM system. The PLM system provides a cooperation platform for collaborative product development. It allows multiple departments and groups to design and develop the same product simultaneously. The PLM system also manages information centrally. It can avoid overlaps and duplications of information storage. It improves the efficiency of the product development and save the unnecessary cost of printing on paper. According to the investigation, the PLM system can save over £200k cost of printing directly every year. The system developers believe that the PLM system can bring over millions pounds benefit every year for improving efficiency. For example, the development of the annual benefit report of the collaborating company is reduced from 9 days to 7 days through the PLM system.

The PLM system is not the only enterprise-level system in the collaborating company. For example, an OA (Office Automation) system is used for daily work, a CAPP (Computer Aided Process Planning) system is used for manufacturing planning, and a financial information system for budget and cost. These systems are mainly independent of each other. There are some communications between these systems, but the communication is not enough. There are more than 10 department-level systems in use. Some departments even have more than one independent system to support their daily work. There is no communication between department-level systems, and there is no communication between department-level systems and enterprise-level systems. This situation leads to a messy management of enterprise information. The collaborating company needs an entire system to manage all information/ knowledge systematically.

There is another limitation of the current IT solution of the collaborating company, i.e., none of the current systems covers the entire business network of the collaborating company. The current IT solution mainly manages the headquarters of the collaborating company. The plants outside the headquarters, the suppliers and contractors are not included in the IT solutions. The communications and information transferring are still not through the IT systems. It still leads to the loss of information and low efficiency when the information is required urgently.

The methodology developed in this project is mainly for better managing product knowledge in the new product development process. Therefore, the PLM system of the collaborating company is the main object for comparing and benchmarking. The PLM system is investigated specifically.

In 2007, the collaborating company launched the project of the PLM. After 3 years, the main system of the PLM is finished. The main aims of the PLM system are managing the process of the product development based on customer requirements, improving cooperation in product design and reducing the lead time of product development. As discussed previously, the PLM system actually improves many areas of the collaborating company, such as saving cost of printing models on paper.

However, from the investigation, there are many limitations of the PLM system in the collaborating company. First of all, although the PLM system considers business requirements, the PLM system implementation has not sufficiently considered end users' requirements. According to the investigation, many interviewees mentioned that the usability of the interfaces and functions are not satisfied. Many functions users wanted are not included in the PLM system. For example, one of the main aims of the PLM system is to provide a design platform for designers. The car-body framework is one of the complex components in the car design. The head of car-body framework centre mentioned that the PLM system cannot store or link necessary functions of the systems and software which are currently being used in the car-body framework centre. Therefore, the car-body framework centre has to abandon the PLM system and use its old system to finish the designs. This situation means that the design of the car-body framework is still done outside of the PLM system. If some others require the design of the car-body framework, the design information will be transferred manually. It leads to more mistakes in the transferring process and low efficiency. There are still many users questioned that the PLM system cannot satisfy their requirements, therefore they have to change their working habits to accommodate the PLM system.

The second limitation of the PLM system is that there is no flexibility in the PLM system implementation. It is also not suitable for SMEs (small and medium enterprises). This limitation leads to the plants outside the headquarters, contractors and suppliers being restricted to the use of the PLM system. The information cannot be delivered in time when the information is required.

The third limitation is that the preparation period of setting up the PLM system is long. The collaborating company used 3 years to finish the PLM system implementation, and it used 2 more years to adjust and generalise the PLM system for the entire company. The cost for implementation, licensing and maintenance is also high. It does not allow many systems and software to be combined or linked either.

Finally, the PLM system is not implemented for managing knowledge. During this project, the company realised and agreed that enterprise knowledge is very important, and the PLM system should have the capability to manage enterprise knowledge, as well as information.

Based on the above discussion, the collaborating company needs a comprehensive system to manage its information/knowledge systematically. The system should consider both business requirements and users' requirements, and also can cover or combine all the current systems in the collaborating company. The new system not only manages information, but also the enterprise knowledge required for product development. Therefore, a requirement driven knowledge management system is needed in the collaborating company.

8.2. Evaluation of the Developed Methodology

According to the guideline of the developed methodology, the system specification of an information system that can manage both information and knowledge is generated. The system specification should be evaluated by knowledge users before it is used for system development or implementation. This evaluation can thus indicate the quality and usefulness of the developed methodology. In this section, the method and results of the evaluation will be discussed.

The method used to evaluate the system specification is to interview and set up meetings with knowledge users and IT support team. In this project, interviews include both telephone interviews and face to face interviews. The meetings mainly focused on one Product Management Department which is used as the case study in this project.

The result of the evaluation is promising. All system specifications are satisfied with knowledge users. Knowledge users can also understand some low prioritised users' requirements that are unnecessary to be implemented in the knowledge management system. The knowledge users commented on the benefits and advantages of the developed methodology, which are summarised below.

First of all, the developed methodology considers both business requirements and knowledge users' requirements. It also provides an easy implementation for the developed methodology. Therefore, the necessary functions of the system are comprehensive, but the functions are not redundant. In other words, system developers and implementers can use the developed methodology to develop a most appropriate knowledge management system based on its own circumstances. Moreover, when the knowledge management system is developed and implemented, the business capability is also considered. It avoids the development of a knowledge management system that the collaborating company cannot afford.

Secondly, the knowledge management system considers both information management and knowledge management in the system. Most manufacturing companies already use several information systems to support their business such as PLM systems and ERP (Enterprise Resource Planning) systems. If a new system is introduced which just manages knowledge, then it will bring some confusion for employees in the company. This situation may lead to an uncertainty in their daily work. This is not a result which the board wants to see.

Thirdly, the developed methodology is easy to understand and use. As discussed previously, the developed methodology consists of a guideline and a “Folder-based EAF Implementation”. System developers and implementers just need understand some concepts of Improved EAF and follow the guideline, and then they can develop and implement their own knowledge management systems.

Fourthly, the developed methodology is suitable for any company. In other words, it is not restricted to the size of the company. As mentioned before, any organisation can be viewed as an “enterprise”. This “enterprise” can be the whole company, a department or a group of the company, any plants, contractors and suppliers.

The knowledge management system specification developed using the proposed methodology provides a satisfied communication vertically and horizontally. One of most popular problems of the collaborating company is the lack of the communication between departments and groups. The knowledge management system allows all knowledge users to communicate simultaneously through distributed design environments, emails, blogs, messages and so on. All kinds of these communications are managed as knowledge in the system, in order to find the relevant knowledge easily.

The developed methodology provides a flexible design of the knowledge management system. As discussed with the IT department, it is possible to allow the knowledge management system to be linked or associated with other existing systems in the collaborating company. The proposed methodology is developed based on the Open Source EAF. Therefore, the costs and licences are free. The maintenance cost is hard to be estimated in the current stage.

To summarise, the knowledge management system can overcome many limitations of PLM systems. One of interviewees clarified one of the possible advantages during the evaluation. Before they used PLM system, the time period of developing annual finance report is 9 days. The current time period of using PLM system is 7 days. The interviewee can assume that if they use the knowledge management system, the developing time period of the annual report can be reduced to 3 days.

There are still some limitations of the developed methodology. The developed methodology is not directly evaluated by system developers (although IT support team in the company are interviewed). One of the most concerns of the interviewees is that the proposed methodology is not mature enough. Therefore, the risk of applying it is high. Also, there are still some unfinished aspects in the proposed methodology. There are still some unknown aspects in the proposed methodology, such as the combination with other systems. These unfinished aspects will be further developed.

8.3. Summary

In this chapter, the IT solution of the collaborating company is reviewed, particularly the PLM system of the collaborating company is assessed. The limitations of the PLM system are discussed. Secondly, the evaluation of the outcome (the system specification of the IT system to be designed) using the proposed methodology is described. The evaluations adopted interviews and meeting discussions as the methods. The result of the evaluation is promising. All the benefits and advantages of the proposed methodology are described and discussed in this chapter. Finally, the limitations of the outcomes and methodology are identified and discussed.

Chapter 9: Conclusions and Further Work

This chapter presents the main conclusions of the PhD project and the recommended areas for further research and development.

9.1. *Conclusions*

During this research project, a formal methodology for the design and development of information systems which can manage enterprise knowledge in new product development as well as information is developed. An automobile company was used to develop and test the methodology. However, the methodology can be applied to other manufacturing companies and general business organisations. A main novel point to be noted is that both business objectives and knowledge user's requirements are used as the main drivers of the knowledge system development.

Through the literature review of current and previous research and development work it has been indentified that there is no formal methodologies for the design and implementation of knowledge management systems considering both business objectives and knowledge user's requirements in new product development applications. The current technologies have different kinds of problems or limitations. The industrial investigations carried out in this project identified that there is a strong requirement for better manage enterprise knowledge, especially knowledge in the new product development process. The research idea has been agreed and approved by the industrial collaborator as targeting one of their strategic priorities.

The developed methodology enhances and extends the capabilities of Enterprise Architecture Frameworks widely used in the information and communication systems industry, and uses the Open Group Architecture Framework (TOGAF) as the basis to develop the methodology, so that the knowledge management system to be developed will be in line with an enterprise's information system. The main advantage of the proposed methodology is that it emphasises knowledge management, whilst existing enterprise architecture frameworks are mainly for information management.

The proposed methodology consists of a guideline (in the form of a flowchart), an improved Enterprise Architecture Framework based on TOGAF, and a folder-based Enterprise Architecture Framework Implementation. The guideline provides a process which can be followed by system developers. The folder-based implementation provides methods and examples of each task and feature in the guideline.

A case study was conducted in an automobile company to verify the developed methodology and its set of tools and guidelines. The case study proved that the manufacturing company can use the methodology to produce the system specifications of their IT systems to include knowledge management capability. The system specification can then be used, either to assess a company's existing information systems and direct its future system development and implementation; or to develop/implement a complete new information system from scratch.

The proposed methodology is compared and benchmarked with current Product Lifecycle Management (PLM) systems and the IT solutions in the collaborating company. The results confirmed that the proposed methodology has capability to develop a comprehensive system for managing both information and knowledge based on business objectives and knowledge user's requirements.

In summary, the main achievements of this research project are:

- A fully investigated knowledge needed in new product development, business process and organisational structure of a typical automotive manufacturing enterprise,
- Identified and classified requirements of different stakeholders and knowledge user's requirements in the new product development process,
- A developed and tested formal methodology for the design and the development of knowledge management systems based on enterprise architecture frameworks,
- A guideline and a folder-based enterprise architecture framework Implementation for enterprise information and knowledge system developers,
- A developed system specifications a knowledge management system for the collaborating company's product development business using the proposed methodology, and
- Results of the verification and evaluation of the developed methodology using the industrial case study, including benefits, limitations and recommended further work.

9.2. *Further* Work

The original purpose of the project is to develop a formal methodology not only for information management, but also for enterprise knowledge management in the new product development domain. In this research project, an automobile company is used for industrial investigation, case study, verification and evaluation. It may lead to some limitations of verification and evaluation. The developed methodology is evaluated indirectly in the current status of this research project. It needs to be evaluated by system developers who actually use it in the further research. Therefore, the proposed methodology should be verified and evaluated in other manufacturing companies making different products and of different business structures, in order to prove the generality of the methodology. As the methodology has the potential to be applied in other business sectors, future tests and further development may include sectors such as aerospace, construction and services.

The case study used Product Management Department as the tested enterprise, and proved that the proposed methodology is appropriate in real industrial environment. The methodology can be applied to an entire enterprise, therefore further tests (and development) should be carried in the context of extended enterprise including suppliers, customers and other business partners.

When using the developed methodology to design a knowledge management system, only the system specifications of the potential knowledge management system are generated. The development (coding, testing, and evaluation) of the knowledge management system is out of the scope of this research project. This can be developed by professional programmers. Another reason for not developing the full system is that the system specifications may be used to assess existing information systems and to recommend further improvement or purchasing of new systems, not just for complete new system development. Even so, further system development work may be carried out to prove that a 'final' result, i.e., a new information system to manage knowledge, can be generated using the system specifications.

This research project mainly focuses on the Architecture Development Method (ADM) of the Enterprise Architecture Framework. Most applications and implementations are based on ADM. The Enterprise Continuum and the Architecture Capability Framework are used as principles and supporting elements. The Enterprise Continuum and Architecture Capability need to be further enhanced in the future work.

The stages of ADM are mainly focused on Preliminary stage, Architecture Vision, Business Architecture, Information Architecture, Technology Architecture, and Opportunities and Solutions. The rest of stages in ADM are not tested in this project, because they are focused on system implementation which is out of the scope of this project. Further work in system implementation will use the full ADM process, i.e., all stages.

References

Abernathy, W.J. and Utterback, J.M. (1978), "*Patterns of Industrial Innovation",* Technology Review, pp. 41-47.

Abramovici, M. and Soeg, O.C. (2002), "*Status and Development Trends of Product Lifecycle Management Systems"*, Ruhr-University Bochum, Chair of IT in Mechanical Engineering (ITM), Germany.

Ambler, S.W., Nalbone, J. and Vizdos, M. (2005), "*Enterprise Unified Process: Extending the Rational Unified Process"*, Prentice Hall.

Alavi, M. and Leidner, D.E. (1999), "Knowledge management systems: issues, challenges and benefits", *Communications of the AIS* (7), pp.1-37.

Bahill, A.T. and Dean, F.F. (2009), "*Handbook of Systems Engineering and Management*", A.P. Sage and W.B. Rouse (Eds.), 2nd edition, John Wiley & Sons, pp. 175-220, 2009.

Bahill, A.T. and Gissing, B. (1998), "Re-evaluating systems engineering concepts using systems thinking", *IEEE Transactions on Systems*, Man and Cybernetics, Part C: Applications and Reviews, Volume 28, Number 4, pp. 516-527.

Baxter, D.I. and Gao, J.X. (2005), "Process based design knowledge reuse through process and product representation", *Proceedings of the 12th CIRP Life Cycle Engineering Seminar (CIRP LCE)* 2005, Grenoble, France April 3-5 2005.

Baxter, D.I., Gao, J., Case, k., Harding, J., Young, B., Cochrane, S. and Dani, S. (2007), "A framework to integrate design knowledge reuse and requirements management in engineering design", *Robot Computer Integration Manufacturing (2007),* doi:10.1016/j.rcim.2007.07.010.

Beck, K. (2001), "*Extreme programming: explained*", 7th edition, Addison-Wesley, USA.

Bender, B., Reinicke, T. and Wunsche, T. (2002), "Application of methods from social sciences in design research", *in: Proceeding of Design 2002, Dubrovnik, Croatia*, 2002.

Bernal, J.D. (1970), "*Science and Industry in the Nineteenth Century*", Bloomington: Indiana University Press.

Berztiss, A. and SYSLAB (2000), "Knowledge and workflow systems", *in: Proceedings of the 11th International Workshop on Database and Expert Systems Applications,* pp. 1102–1106.

Blackchard, B.S. and Fabrycky, W.J. (1990), "*System Engineering and Analysis*", 1st Edition, Prentice Hall.

Briggs, H.C. (2006), "Knowledge Management In The Engineering Design Environment", *the 47th IAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, Newport, Rhode Island, 1 - 4 May 2006.*

Brown, L.R. (2003), *"Eco-Economy: Building an Economy for the Earth",* Earthscan.

Burrows, G.R., Drummond, D.L. and Martinsons, M.G. (2005), "Knowledge Management In China," *Communications of the ACM*, Vol 48, No 4, pp73-76.

Bush, A.A. and Tiwana, A. (2005), "Designing Sticky Knowledge Networks," *Communications of the ACM*, Vol. 48, No. 5, May 2005.

Chapman, W.L., Bahill, A.T. and Wymore, W. (1992), "*Engineering Modeling and Design*", CRC Press, Boca Raton.

Chatzoglou, P.D. (1997), “Factors affecting completion of the requirements capture stage of projects with different characteristics”, *Information and Software Technology*, Volume 39, Issue 9, 1997, pp. 627-640.

Chen, J.Q., Lee, T.E., Zhang, R. and Zhang, Y.J. (2003), “System Requirements for Organizational Learning,” *Communications of the ACM*, Vol 46, No 12, Dec 2003, pp 73-78.

Chen, P.P.S. (1976), “The entity-relationship model—toward a unified view of data”, *ACM Transactions on Database Systems (TODS),* Volume 1, Issue 1 Pages: 9 – 36.

Choi, B., Poon, S.K., and Davis, J.G. (2008), “Effects of knowledge management strategy on organizational performance: A complementarity theory-based approach”, *Omega*, Volume 36, Issue 2, Pages 235-251.

Choi, I., Jung, J. and Song, M. (2004), “An integrated framework for process knowledge management”, *International Journal of Innovation and Learning, Vol.* 1(4), (2004), pp.399–408.

Chua, A. (2004), “Knowledge management system architecture: a bridge between KM consultants and technologists”, *International Journal of Information Management*, 24(2004), pp.87-98.

CIMdata, “Product Lifecycle Management (PLM) Definition”, *Available: http://www.cimdata.com/plm/definition.html*, accessed: 01-05-2011.

CMS (Centers for Medicare and Medicaid Services of U.S. Department of Health and Human Services), 2005, “Selecting a Development Approach”, available at *http://www.cms.gov/SystemLifecycleFramework/Downloads/SelectingDevelopmentApproach.pdf*, accessed on 21-04-2011.

CIO Council (2001), "*A Practical Guide to Federal Enterprise Architecture*", the CIO Council, Version 1.0.

Crawford, C.M. (1979), "New product failure rates - facts and fallacies", *Research Management* **22**(5), pp. 9-13.

Court, A.W. (1997), "The relationship between information and personal knowledge in new product development". *International Journal of Information Management* **17**(2), pp. 128-138.

Dallas, M. (2006), "*Value and risk management: a guide to best practice*", Oxford: Blackwell.

Davenport, T.H. and Prusak, L. (1998), "*Working knowledge: how organizations manage what they know"*, 1st edition, Boston, MA: Harvard Business School.

Davis, M. and Zowghi, D. (2006), "Good requirements practices are neither necessary nor sufficient". *Require Eng 2006*, 11:1-3.

DoD Architecture Framework Working Group (2003) "*DoD Architecture Framework: Volume I: Definitions and guidelines",* Architecture Framework Working Group.

Drucker, P. F. (1998), "*The Coming of the New Organization*", Harvard Business Review on Knowledge Management, Harvard Business School Press.

Eden, A and Kazman, R (2003) "Architecture, Design, Implementation", *International Conference for Software Engineering*, pp. 149 – 159.

Emam, K.E. and Madhavji, N.H. (1995), "A Field Study of Requirements Engineering Practices in Information Systems Development", *Proceedings of the Second IEEE Int. Symp on Requirements Engineering*, York, UK, pp. 68–80

European Committee for Standardization (2010), "Knowledge Management", *Available at* *http://www.cen.eu/CEN/sectors/sectors/isss/cen%20workshop%20agreements/Pages/knowledge%20management.aspx*, Accessed on: 02-05-2011.

Federal Enterprise Architecture Program Management Office (2006), "FEA Consolidated Reference Model Document V2.1", published by *Federal Enterprise Architecture Program Management Office.*

Fernandez, L.F. and Moreno, A.V., (2004), "An introduction to UML profiles", The European Journal for Informatics Professional, Vol. V. No. 2.pp. 6.

Firesmith, D. (2004), "Prioritizing Requirements", *Journal of Object Technology*, vol. 3, no. 8, September-October 2004, pp. 35-47.

Gartner (2009), "*Business Intelligence Submit 2009*", Time, Page7

Gido, J. and Clements, J.P. (2004), "*Successful Project Management*", 2nd Edition, South-Western College Publishing

Gold, A.H., Malhotra, A. and Segars (2001), A.H., "Knowledge management: An organizational capabilities perspective", *Journal of Management Information Systems* **18** (2001) (1), pp. 185–214.

Haag, S., McCubbrey, D.J. and Cummings, M. (2001), "*Management and Information Systems for the Information Age*", 3rd Edition, McGraw-Hill Higher Education

Hughes, T.P. (1989), "*American Genesis: A History of the American Genius for Invention*", Penguin Books.

Ichijo, K. and Kohlbacher, F., (2008), "Tapping tacit local knowledge in emerging markets – the Toyota way", *Knowledge Management Research and Practice* (2008) b, pp173-186

IDEF Web site, "IDEF0 Functional Modelling Method", *Available at http://www .idef.com/idef0.htm*, accessed at 02-05-2007.

James, G.A., Handler, R.A., Lapkin, A. and Gall, N (2005), "*Gartner Enterprise Architecture Framework: Evolution 2005*", Gartner Research, ID Number:G00130855

Jung, J., Choi, I., Song, M. (2006), "An integration architecture for knowledge management systems and business process management systems", *Computers in Industry* 58 (2007) pp. 21–34

Kalpic, B., and Bernus, P. (2002), "Business process modelling in industry—the powerful tool in enterprise management*", Computers in Industry,* Volume 47, Issue 3, Pages 299-318.

Karlsson, J., Wohlin, C. and Regnell, B. (1997), "An evaluation of methods for prioritizing software requirements", *Elsevier, New York*, 7 Feb. 1997, pp 939-947

Katzenbach, J.R., and Douglas K.S. (1993), "*the Wisdom of Teams: Creating the High-Performance Organisation*", Harvard Business School Press, Boston

Keith L.W. and Vincent J.B.Jr. (2005) "Applying the Design Structure Matrix (DSM) Technique to NASA Organizational Design", *NASA APPL 2nd Project Management Challenge Conference*

Kerzner, H., "Project Management: a Systems Approach to Planning, Scheduling, and Controlling", Van Nostrand Reinhold, New York, 1995.

Khurana, A., and Rosenthal, S.R.(1998) , "Towards holistic "front ends" in new product development", *Journal of Product Innovation Management*, Volume 15, Issue 1, Pages 57-74.

Kidder, T. (1981), "*the Soul of a New Machine*", Avon Books, New York

Knight, L.V, Steinbach, T.A., and Kellen, V (2003), "System Development Methodologies for Web-Enabled E-Business: A Customization Framework", *Architectural Issues of Web-Enabled Eletronic Business,* Idea Group Inc., pp213-226.

Ko, Y., Park, S., Seo, J. and Choi S. (2007), "Using classification techniques for informal requirements in the requirements analysis-supporting system", *Information and Software Technology*, 49 (2007), pp.1128 –1140.

Kuang L., and Gao, J. (2006), "A framework to integrate manufacturing information system", *proceeding of the CORP International conference on digital enterprise technologies (DET2006), 18-20 September 2006, Sebubal Portugal. ISBN:972-99824-1-4, ed.* Pedro F Chunha and Paul Maropoulos.

Kumar, R. (2011), "*Research Methodology: A Step-by-Step Guide for Beginners*", 3rd edition, Sage Publication Ltd.

Lai, J., Fan, Y. (2002), "Workflow and knowledge management: approaching anintegration", in: *Proceedings of the 1st International Conference on Engineeringand Deployment of Cooperative Information Systems*, Vol. 2480of Lecture Note in Computer Science, Springer-Verlag, Berlin, pp.16–29.

Lee, H. and Choi, B., (2003), "Knowledge management enablers, process, and organisational performance: an integrative view and empirical examination", *Journal of Management Informational System*, Vol.20, No.1, pp.179-228

Leedy, P.D. (1989), "*Practical research: planning and design: a basic text for all courses in research methodology"*, 4th edition. New York, United States of America: Macmillan Publishing Company.

Leffingwell, D. and Widrig, D. (2003), "*Managing Software Requirements: A Use Case Approach*", 2nd edition, Addison-Wesley, USA.

Levin, M.S. (2006), "*Composite Systems Decisions*", Springer, New York.

Liebowitz, J. (1999), "*Knowledge management handbook"*, Florida, United States of America: CRC Press Inc.

Liao, S.H. (2003), "Knowledge management technologies and applications – literature review from 1995 to 2002", *Expert Systems with Applications* 25 (2003), pp 155 – 164

Mantoux, P. (1961), "*The industrial revolution in the eighteenth century : an outline of the beginnings of the modern factory system in England"*, London: Jonathan Cape.

Marbán, O., Menasalvas, E., and Fernández-Baizán, C. (2008), "A cost model to estimate the effort of data mining projects (DMCoMo)", *Information Systems*, Volume 33, Issue 1, pp. 133-150.

Maurer, F., Dellen, B. (1998), "A concept for an internet-based process-orientedknowledge management environment", in: *Proceedings of the 11th work- shop on Knowledge Acquisition Workshop (KAW'98), Alberta, Canada*,18–23, pp. 18–23.

Maurer, F., Holz, H. (1999), "Process oriented knowledge management for learning software organizations", in: *Proceedings of the 12th Knowledge Acquisition Workshop (KAW'99), Banff, Canada.*

Medvidovic, N. and R. Taylor (2000), "A Classification and Comparison Framework for Software Architecture Description Languages", *IEEE, Transaction on Software Engineering*, Vol. 26, No. 1

Meredith, D.D., Wong, K.W., Woodhead, R.W. and Wortman, R.H. (1985), "*Design Planning of Engineering System*", Prentice-Hall, Englewood Cliffs, NJ

Meredith, J.R. and Mantel, S.J. Jr. (2006), "*Project Management- A Managerial Approach*", 6th Edition, John Wiley & Sons, Inc

Miller, P. (1998), "*Mobilising the Power of What you know*", London: Random House.

Ming, X.G., Tan, J.Q., Wang, X.H., Li, S.N., Lu, W.F., Peng, Q.J., and Ma, Y.S. (2007), "Collaborative process planning and manufacturing in product lifecycle management", *Computers in Industry*, 59 (2008) 154-166

Nilsson, P. and Fagerström, B. (2006), "Managing stakeholder requirements in a product modelling system", *Computers in Industry*, Volume 57, Issue 2, Pages 167-177.

Nonaka, I and Treece, D. (2001), "*Managing Industrial Knowledge*", Sage, London

Noran, O. (2003), "An analysis of the Zackman framework for enterprise architecture from the GERAM perspective", *Annual Reviews in Control*, Volume 27, Issue 2, Pages 163-183

Norcliffe, G. (1997), "Popeism and Fordism: Examining the Roots of Mass Production", *Regional Studies*, 31(3), pp.267-280.

O'Marah, K. and Myer, B. (2002), "*The product lifecycle management applications reports 2001-2006*", AMR Research

Pattron, D., (2000), "*Fundamentals of Scientific Research*", New York: Scientific Publishers.

Park, H and Cutkosky, M.R. (1999), "Framework for Modelling Dependencies in Collaborative Engineering Processes"*, Research in Engineering Design* 11:84-102

PLM Technology Guide, "What is PLM", *available at:* http://plmtechnologyguide.com/site/?page_id=435, accessed: 02-05-2011

Polanyi, M. (1966), "*The Tacit Dimension"*, Doubleday, Garden City, N.Y.

Powell, T.C and Dent-Micallef, A. (1997), "Information technology as competitive advantage: the role of human, business and technology resource", *Strategic management Journal*, Vol. 18:5, pp.375-405

Project Management Institute (1996), "*Project Manager Body of Knowledge (PMBOK)*", Project Management Institute

Richardson, S.M., Courtney, J.F. and Haynes, J.D., (2006), "Theoretical principles for knowledge management system design: application to pediatric bipolar disorder", *Decision Support Systems* 42(3), (2006), pp. 1321 – 1337

Rodgers, P. A. and Clarkson, P. John, (1998b), Knowledge usage in new product development (NPD): *International Conference on Design and Technology Educational Research and Curriculum Development (IDATER 98)*, pp. 252-258.

Ross, D.T. (1977), "Structure Analysis (SA): A language for communicating ideas". *IEEE Trains SoftW Eng 3(1)*: 16-34.

Ratchev, S., Urwin, E., Muller, D., Pawar, K.S. and Moulek, I. (2003), "Knowledge based requirement engineering for one-of-a-kind complex systems", *Knowledge-Based Systems*, Volume 16, Issue 1, Pages 1-5.

Riege, A. (2005), Three-dozen knowledge-sharing barriers managers must consider. *Journal of Knowledge Management*, 9(3), 18-35.

Rosenberg, D. and Scott, K., (1999), "Use case driven object modelling with UML: a practical approach", *Addison-Wesley Longman Publishing Co., Inc*, Boston, MA, USA.

Ruggles, R. (1998), "The state of the notion: Knowledge management in practice", *California Management Review*, 40(3), 80–89.

Saaty, T. (1980), "*The Analytic Hierarchy Process: Planning, Priority Setting, Resource, Allocation*", McGraw-Hill, New York.

Saint-Onge, H. (1996) "Tacit Knowledge: The Key to the Structure Alignment of Intellectual Capital, Strategy and Leadership", March/April pp10-14

Sarvary, M. (1999), "Knowledge management and competition in the consulting industry", *California Management Review 1999*; 41(2):95–107.

Sessions, R. (2007), "A Comparison of the Top Four Enterprise-Architecture Methodologies", *ObjectWatch*, Inc

Siemens PLM system, "Defining Siemens PLM", http://www.plm.automation.siemens.com/en_gb/plm/definition/, accessed on 13 – May-2011.

Sharma, R. and Gao, J.X. (2006), "A knowledge-based manufacturing and cost evaluation system for product design/re-design", *The International Journal of Advanced Manufacturing.*

Sparks, G. (2000), "An introduction to modelling software system using the Unified Modelling Language – Business process model", Enterprise Architect, available at http://www.imamu.edu.sa/DContent/IT_Topics/An%20Introduction%20to%20UML.pdf, accessed on 05-10-2011

Srinivasan, V. (2008), "An integration framework for product lifecycle management", *Computer-Aided Design*, doi:10.1016/j.cad.2008.12.001

Sommerville, I. (2006), "*Software Engineering*", *Addison-Wesley.*

Steward, D.V. (1981), "The design structure system: A method for managing the design of complex systems". *IEEE Trans Eng Manage,* 28:71-74

Steward, D.V., (1981b), "*Systems Analysis and Management: Structure, Strategy and Design*", Petrocelli Books, New York

Stokes, M. (2001), "*Managing Engineering Knowledge—MOKA: Methodology for Knowledge Based Engineering Applications*", Professional Engineering Publishing Limited

Sudarsan, R. Fenves, S.J., Sriram, R.D. and Wang, F. (2005), "A product information modelling framework for product lifecycle management", *Computer-Aided Design*, 37(2005) 1399-1411

Tang, A., Han.,J. and Chen, P. (2004), "A Comparative Analysis of Architecture Frameworks", *SUTIT*-TR2004.01

Tang, X. and Yun, H. (2007), "Data Model for quality in product lifecycle", Computers in Industry, 59 (2008) 167-179

The Object Management Group (2011), "Introduction to OMG's Unified Modelling Language (UML)", available at http://www.omg.org/gettingstarted/what_is_uml.htm, accessed on 05-10-2011

The Open Group (2007), "The Open Group Architecture Framework version 8.1.1 Enterprise Edition", The Open Group

The Open Group (2009), "The Open Group Architecture Framework version 9, Enterprise Edition", The Open Group, 2009 available at http://www.opengroup.org/architecture/togaf9-doc/arch/ (accessed on 02-05-2011)

Toften, K. and Olsen, S.O. (2003), "Export market information use, organizational knowledge, and firm performance: A conceptual framework", *International Marketing Review,* 20(1), 95-110.

Ulrich, K.T. and Eppinger, S.D. (2011), "*Product Design and Development"*, 5th edition, McGraw-Hill*,* New York

Utterback, J.M. and Abernathy, W.J., "A Dynamic Model of Process and Product Innovation", *Omega*, 1975, 3(6), pp.639-56

Vestal, W. (2005), "Making sense out of KM costs," *KMWorld*, Volume 14, Number 7, July/August 2005.

Wilson, P. (1987), "Information modelling", *IEEE Computer Graphics and Applications* 7(12), 65-67

Woodman, K.L. and Bilardo, V.J. Jr. (2005), "Applying the Design Structure Matrix (DSM) Technique to NASA Organizational Design", *NASA APPL 2nd Project Management Challenge Conference*

Wyatt, J.C., (2001), "Management of explicit and tacit knowledge", *J, Roy, Soc, Med.*, 94, pp, 6-9

Zackman. J. (2003), The Zackman framework: a primer for enterprise engineering and manufacturing, available at http://www.zachmanframeworkassociates.com/ (accessed 02-05-2011)

Zdrahal, Z., Mulholland, P., Valasek, M. and Bernardi, A. (2007), "Worlds and transformations: Supporting the sharing and reuse of engineering design knowledge", *International Journal of Human-Computer Studies*, *Volume 65, Issue 12*, *Pages 959-982.*

Appendix A: The Questionnaires for the Industrial Investigation in the Collaborating Company

Questionnaire 1 for the Industrial Investigation in the Collaborating Company

Name: Xing

Date: 02-08-2008

Position: Head of the Planning Department

- General Questions
  - What type of product do you produce?

    Tractors, trucks, platform trucks, buses and sports utility vehicles (SUV)

  - Could you describe your work scope and responsibility?

    Planning the whole product development process including production line, budget, cost, and so on

  - How many people are there in your team?

    Two groups: the Planning Group and Concept Development Group, totally, 57 people

- What are the business objectives of a new product development in your company?

  There are many different objectives when we start to set up the main achievement of a new development of product, such as cost saving, market share increasing, benefit increasing, more customer, and so on

- Is your new product development requirements driven? If yes, please describe how requirement drive the development process.

  Yes, it is. For example, the coach we product based on customer's requirements. Customers can select the engines, inner decoration, number of seats and so on based on a template we provided to them.

- How does your company capture, specify and use the various requirements in new product development? Such as customer requirements, business requirements, design requirements and engineering requirements.

  We have a mature product development process which is the business process. It includes requirements conversion from customer requirements to final product requirements.

- Is there any knowledge based engineering in your company? If yes, please describe how to capture, store, share and reuse knowledge in the engineering department.

  No, there is not. But we have already realised that an enterprise system should help us to gain many advantages from many aspects, such as more effectiveness, saving cost, improve product development period, reduce risks and failures.

- Are there any enterprise guidelines or processes as a standard to guarantee the company achieve the business goals and objectives? If yes, please describe it with its architecture. E.g. TOGAF

  No, we do not have. The guideline and full perspective of the enterprise is in boss's head.

- How does your company keep product data? Are there any descriptions of each data in database? E.g. drawing, CAD Data and etc.

  Different departments keep their data in different ways. My department has an individual database to keep the processes, reports, models. However, not all them are kept in the database, a large number of them are still on the paper.

- Is there any KM or IT system in your company? If yes, Can you please introduce the system you use?

  Not specifically, we are developing a PLM system to help product development, but it is still developing and the progress is slow.

- What kind of ideal KM system do you need based on your position or your team?

  An ideal KM system should manage the whole enterprise like one system. People can use the KM system to manage their daily work. Also we can share knowledge and communicate in the system. The most important point is that the KM system should improve the effectiveness and efficiency.

- Are there any special requirements of your team for the system?

  We need system which can plan the whole process of product development. Also we need a template for development of product concept. It can store all knowledge in system, we can use it conveniently.

- What kind of data do you usually get from the common drive?

  Processes, models, product concept report

- Could you simply describe the structure of the common drive? Is it easy for you to get documents following this structure?

  It is just like a computer drive. We need to remember where we keep the knowledge. If not, it will take long time to find it.

- Are you able to find all documents you need from the Intranet? If not, what do you think should be added to it?

  Barely, it is hard to share knowledge in the company. We need some places to find what we need when we need them.

- What kind of data do you usually use? Where do they come from originally?

Processes, models, reports, if there are in the department, we just find ourselves. If not, we need to talk to relevant people to get it.

- How long does it take to get the exact documents that you want from the Intranet usually?

  If it is not in the department, we ask somebody to pass that document to us. It depends on where is the relevant person. It may take days or hours.

- What types of documents do you usually require or share, e.g. Word, Excel, PowerPoint, CAD?

  CAD models, pictures, Microsoft document. Processes and models on paper.

- Are you happy with the authorization of documents given to you and others?

  Definitely, some information is only for inside of department.

Questionnaire 2 for the Industrial Investigation in the Collaborating Company

Content includes:

- Verify the outputs based the last investigation
  - Organisational structure
  - Business process or product lifecycle – ask to verify the detailed process with the current situation
  - Product manager policy
  - Knowledge user's requirements [select different knowledge users, i.e., their 'roles', and then their requirements on 'knowledge' and 'functions of information systems to manage the knowledge']

- New investigation or data collection requirements
  - An example of a system – car door of one or two product(s)
    - Fully understand the structure of the system
    - Process of design

- Conditions or product requirements included in the systems
- Functions of the example system
- Details of the example system such as data (do not need the parameters), suppliers, diagrams for components, roles and departments [and groups and persons] involved
- The detailed and system-level design
- What kinds of [information system] Supports you needed (not only the information system support, but also functional supports with other departments)
- how to plan this system
- Product concept development applied or implemented in this system
- Questions:
  - Could you describe the details of structure of this car door?

    It is confidential. I just give you a brief idea. We need to think many aspects when we design it, such as size, cost, safety, scheme and so on.

  - Could you describe the design process of this car door

    Market investigation → the popular sculpt in the market→ our creation → sculpt definition → 2D Drawings → several schemes → production analysis → board decision → 3D drawings → details design → models creation → models analysis → model prototype→ testing → evaluation → digitalisation→ sample of product

  - When you start to design the car door, what kinds of conditions and requirements you need to concern and involve in the design process?

    We consider market requirement firstly. We also consider policies and regulation, cost , capability of production, quality, budget and so on.

  - What functions are necessary to contains in the car door? Are there any functions you have to give up when you developed the car door? Why?

    Performance, size, safety and so on

  - What kinds of data you need to concern and involve in the car door design? (Product requirements are important, conversion and inputs outputs!)

Size of the body, NVH (noise, vibration, and harshness), materials, quality, weight, impaction, and so on

- When you design a car door, what is your input and output for designing?

  Inputs: concept development report, market requirement, quality objective, competitor, cost evaluation, budget, design objectives

  Outputs: design models, 2D graphs, product functions, quality specification, details of the components, technology requirements

- When you design the car door, what kinds of roles, department and suppliers need to be involved?

  Most all departments are involved, especially Finance department, quality group, suppliers (including inner and outside decoration, windows, light), engine group, testing group

- When you develop the product, what kinds of supports you need?

  KM database, specially needs in-house product knowledge and the data from competitors.

- How do you plan the process of the product development?

  Same as the second question

- How do you apply and implement Product concept development in this system?
  Same as the design process

➢ KM system (also need to ask in example system)

NOTICE: Need to clearly classify types of knowledge, i.e., structured, un-structured and semi-structured, and also explicit tacit knowledge. This classification may only be used for 'the knowledge to carry out the tasks'.

-
- Knowledge user's requirements needed– what knowledge is needed in this system as a knowledge system user (When talking about the KM needs, 3 levels need to ask (1) organisation requirements and product requirements, (2) what information and knowledge needed to carry out the tasks; and (3) information system requirements for managing the knowledge and information)
- Knowledge user's requirements needed – how to manage knowledge or information in current situation of the example system, any limits in the current situation, any more requirements need to be considered in KM system
- Investigation of the functional requirements in the current KM system and what functions you needs in the ideal KM system (based on departments and different roles in the product process, need to find some example base on department and roles)
- Where do you store knowledge currently? (if possible, need to finds out knowledge flow)
- PLM or ERP in the company? Discuss it with the example system
- Questions：
  - Do you have any research or application of PLM or ERP, if you have, could you please describes its details? If not, do you have any similar system or application in your company?

    PLM system mainly is used for design and development product. Designers can use it to design models, processes, components directly in the system. However, it can not satisfy all design requirements.

  - What knowledge is needed in this system as a knowledge system user？ (when talking about the KM needs, 3 levels need to ask organisation requirements, product requirements and KM system requirements)

    Paperless works, process optimisation, modelling process, digitalisation, can cover most of daily work

  - How to manage knowledge or information in current situation of the example system, any limits in the current situation, any more requirements need to be considered in KM system？

Sub-systems in most departments, OA system, PLM system, paper, meeting, conference, telephones, and etc

- How can you store your knowledge and information? Diagram, text, paper and pen, computer files?

  Paper, computer drives, OA system, PLM system, human heads.

- How can you communicate and share knowledge with other departments or roles?

  OA system, PLM system, telephone, meeting, emails, message

  Share some knowledge, but not too much

- Concept development and Planning

  - What are the outputs of the concept development stage? Drawings? Models? Where do you store them？

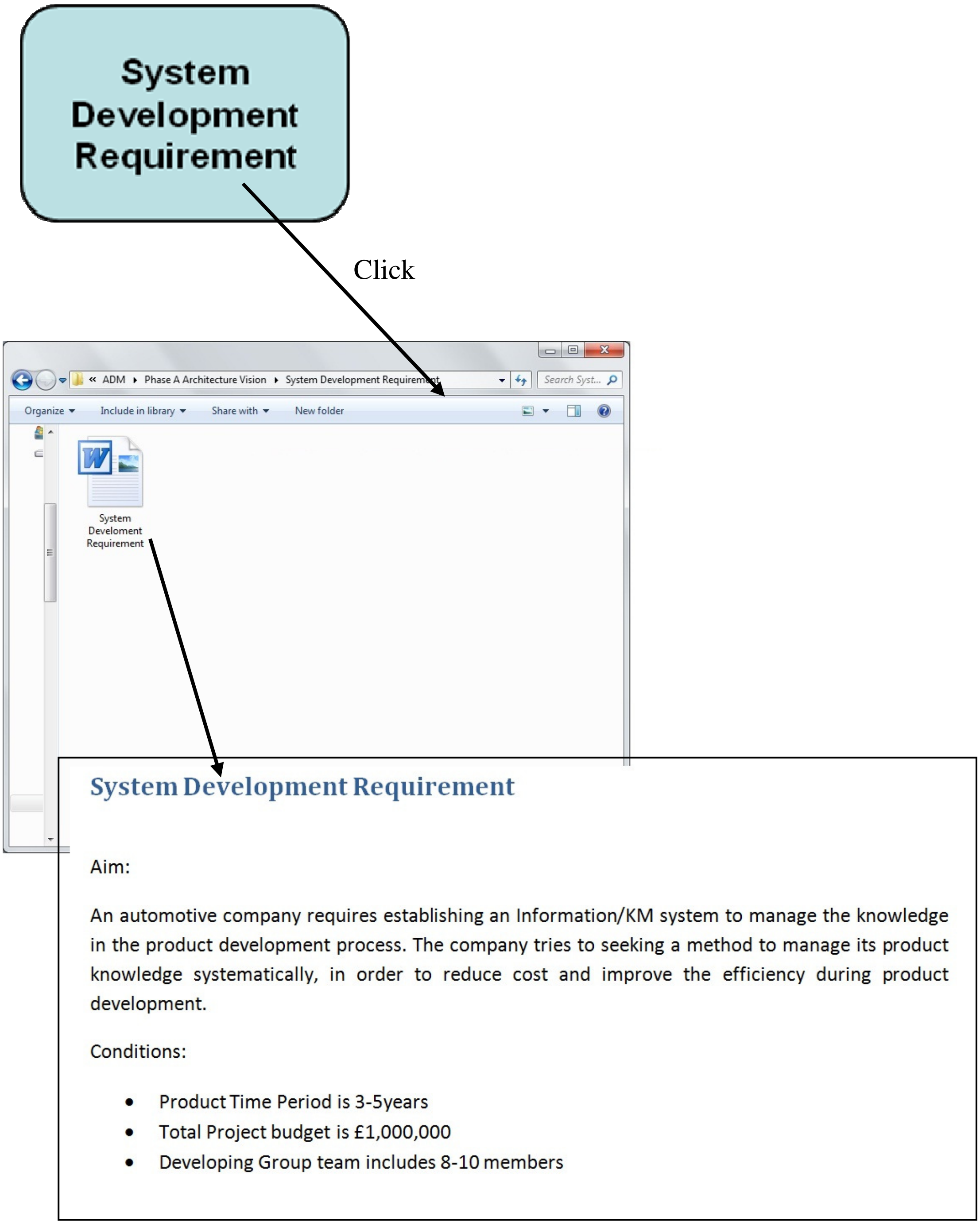
    Concept development report including drawings, roles involved, inputs,, outputs of each stage, draft models.

- **Possible departments involved**
  - Product Management Department
  - Design Department
    - The relevant groups for the design and development
    - Car-body Framework
    - Product Concept Development Related

Appendix B: Screenshots showing "Folder-based EAF Implementation" with The Guideline

This appendix represents the features (inputs and outputs) of the Guideline. Also it is represents the links between the Guideline and the "Folder-based EAF Implementation".

This screenshot represents that system developers apply the “System Development Requirement” Feature of the Guideline in the “Folder-based Implementation” with examples, when they need to know the contents in the Folder-Based EAF Implementation.


System Development Requirement
Click
« ADM ▸ Phase A Architecture Vision ▸ System Development Requirement
Search Syst...
Organize
Include in library
Share with
New folder
System Develoment Requirement

System Development Requirement

Aim:

An automotive company requires establishing an Information/KM system to manage the knowledge in the product development process. The company tries to seeking a method to manage its product knowledge systematically, in order to reduce cost and improve the efficiency during product development.

Conditions:

- Product Time Period is 3-5years
- Total Project budget is £1,000,000
- Developing Group team includes 8-10 members

Figure B.1 – System Requirement Feature implemented in Folder-based Implementation

This screenshot represents that system developers apply the "Semi-Structure Questionnaire" Feature of the Guideline in the "Folder-based Implementation" with examples, when they need to know the contents in the Folder-Based EAF Implementation.

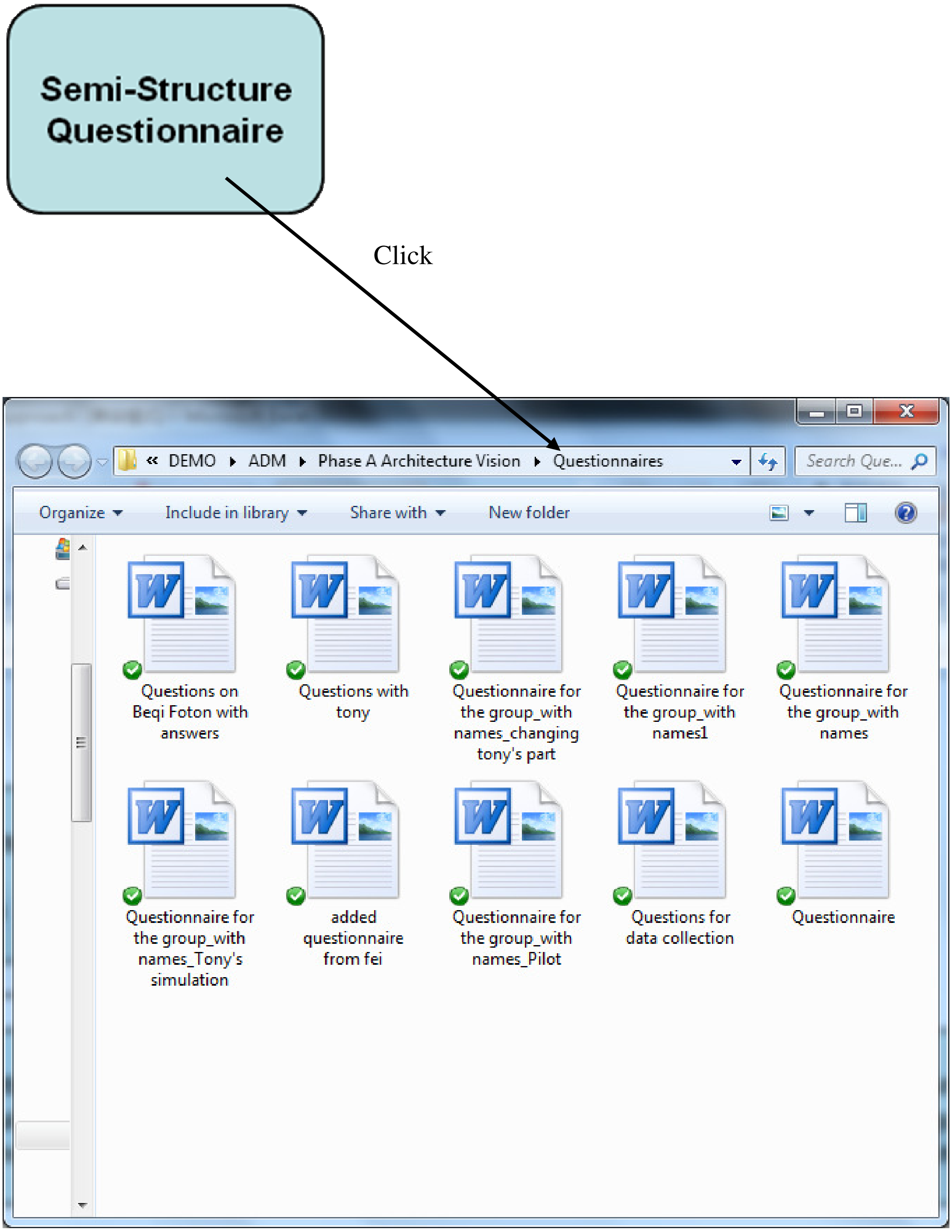
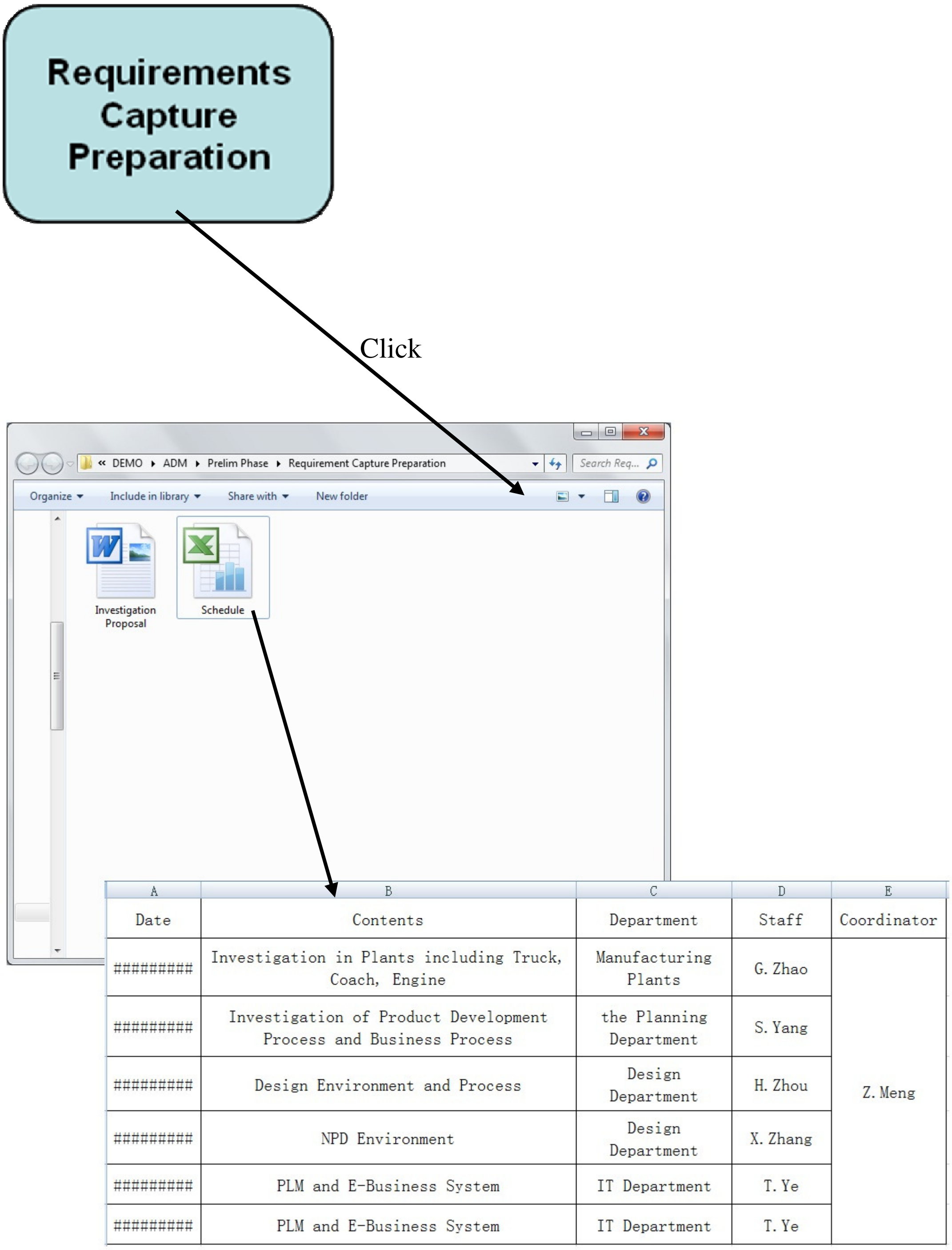

Semi-Structure Questionnaire
Click
« DEMO ▸ ADM ▸ Phase A Architecture Vision ▸ Questionnaires
Search Que...
Organize ▾
Include in library ▾
Share with ▾
New folder
Questions on Beqi Foton with answers
Questions with tony
Questionnaire for the group_with names_changing tony's part
Questionnaire for the group_with names1
Questionnaire for the group_with names
Questionnaire for the group_with names_Tony's simulation
added questionnaire from fei
Questionnaire for the group_with names_Pilot
Questions for data collection
Questionnaire

Figure B.2 – Semi-Structure Questionnaire Feature implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Requirements Capture Preparation” Feature of the Guideline in the “Folder-based Implementation” with examples, when they need to know the contents in the Folder-Based EAF Implementation.



A	B	C	D	E
Date	Contents	Department	Staff	Coordinator
#########	Investigation in Plants including Truck, Coach, Engine	Manufacturing Plants	G. Zhao	Z. Meng
#########	Investigation of Product Development Process and Business Process	the Planning Department	S. Yang	
#########	Design Environment and Process	Design Department	H. Zhou	
#########	NPD Environment	Design Department	X. Zhang	
#########	PLM and E-Business System	IT Department	T. Ye	
#########	PLM and E-Business System	IT Department	T. Ye	

Figure B.3 – Requirements Capture Preparation Feature implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Applying EAF Principle” Feature of the Guideline in the “Folder-based Implementation” with examples, when they need to know the contents in the Folder-Based EAF Implementation.

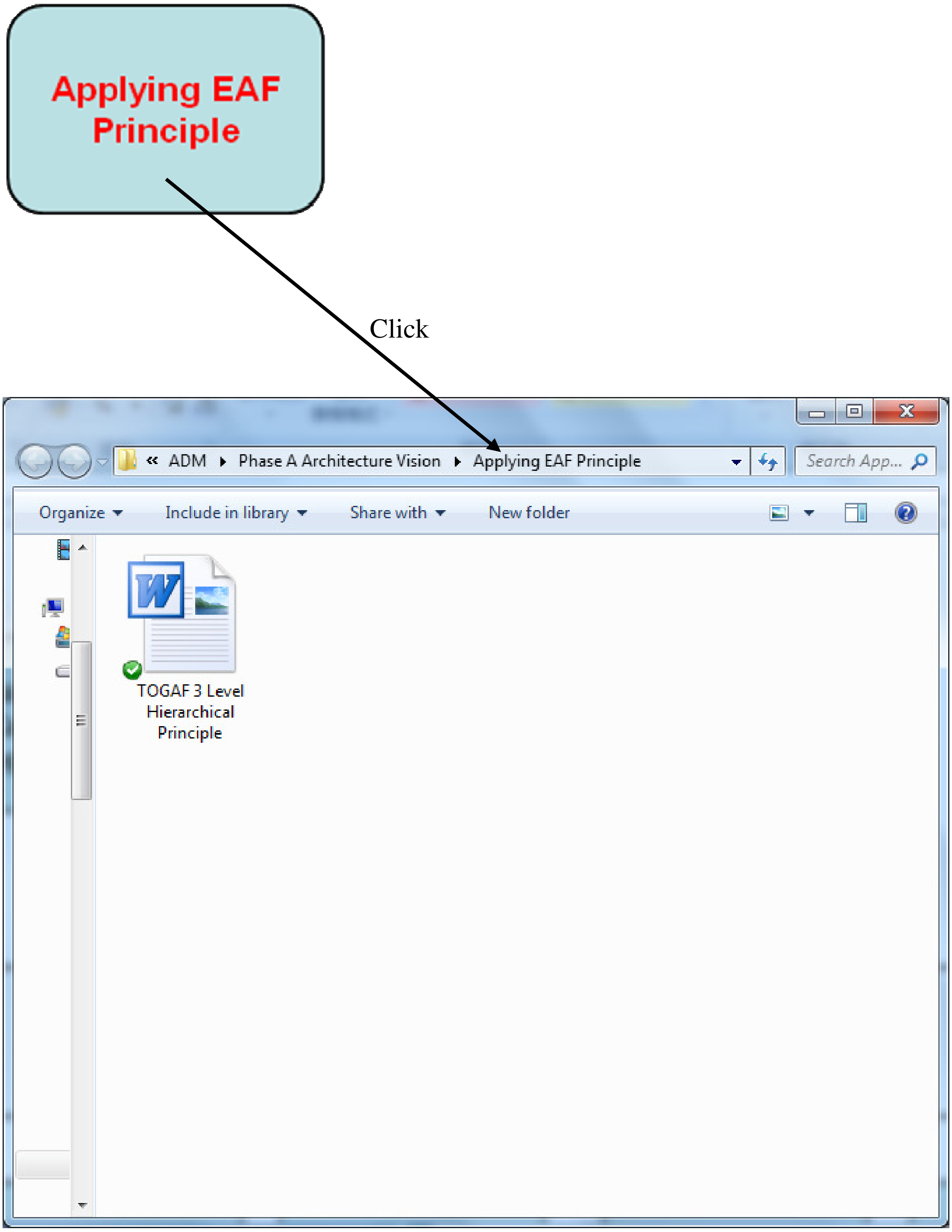

Applying EAF Principle
Click
« ADM ▸ Phase A Architecture Vision ▸ Applying EAF Principle
Search App...
Organize ▾
Include in library ▾
Share with ▾
New folder
TOGAF 3 Level Hierarchical Principle

Figure B.4 – Applying Enterprise Architecture Framework Principle Feature Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Initially Captured User’s Requirements & Knowledge” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix E and Appendix F), when they need to know the contents in the Folder-Based EAF Implementation.

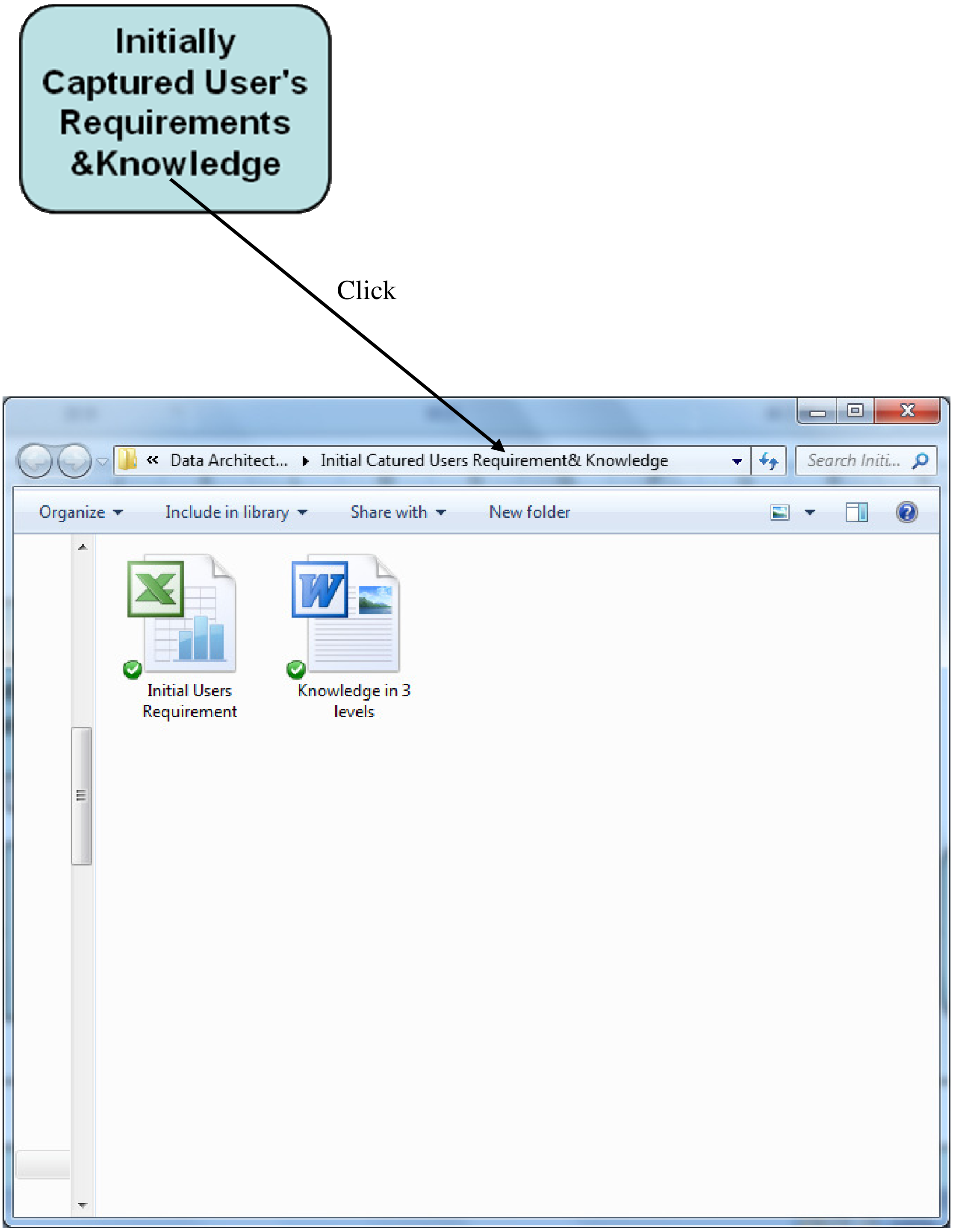
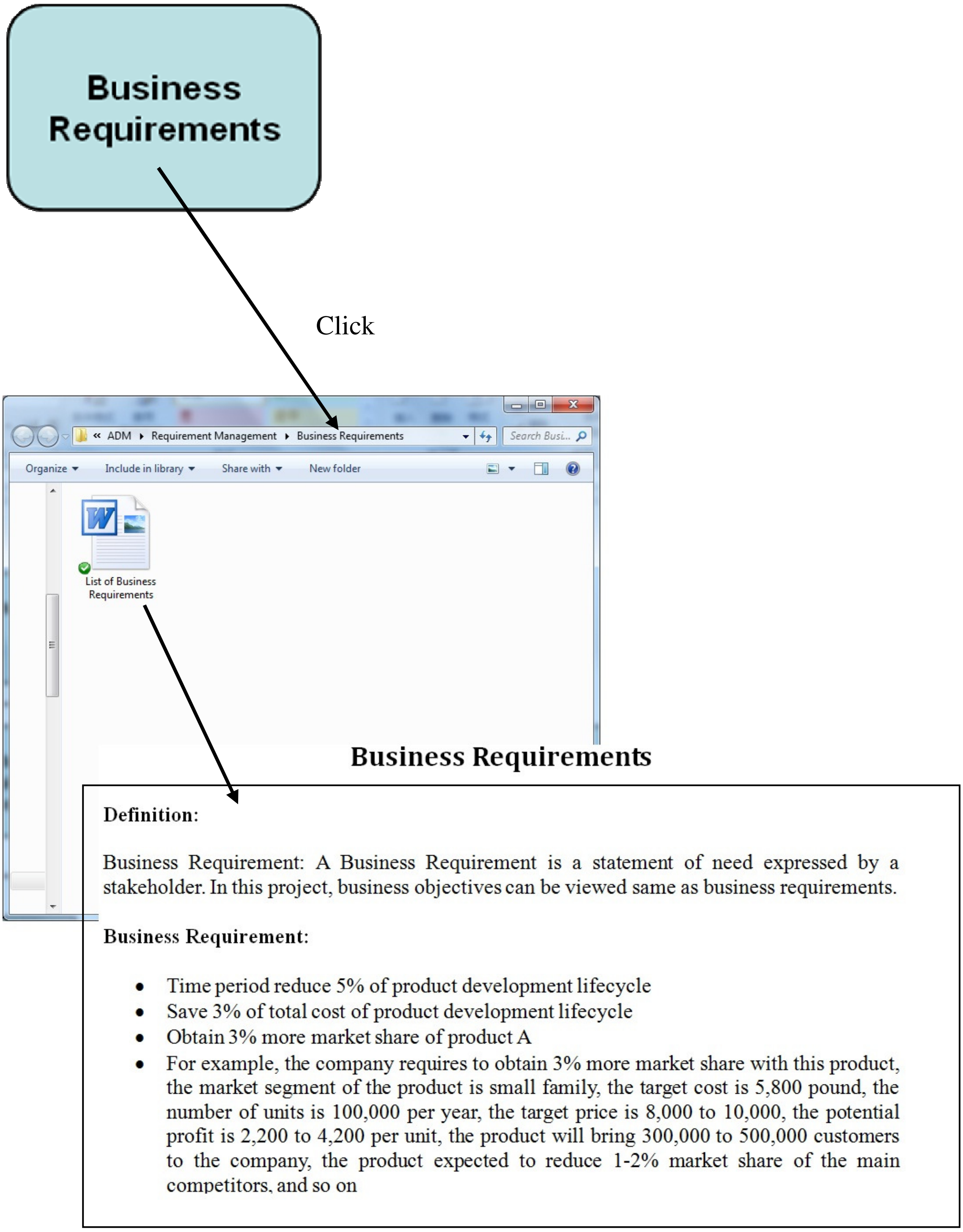

Initially Captured User's Requirements &Knowledge
Click
« Data Architect... ▸ Initial Catured Users Requirement& Knowledge
Search Initi...
Organize ▾
Include in library ▾
Share with ▾
New folder
Initial Users Requirement
Knowledge in 3 levels

Figure B.5 – Initially Captured Users’ Requirements and Knowledge Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Business Requirements” Feature of the Guideline in the “Folder-based Implementation” with examples, when they need to know the contents in the Folder-Based EAF Implementation.



Business Requirements

Definition:

Business Requirement: A Business Requirement is a statement of need expressed by a stakeholder. In this project, business objectives can be viewed same as business requirements.

Business Requirement:

- Time period reduce 5% of product development lifecycle
- Save 3% of total cost of product development lifecycle
- Obtain 3% more market share of product A
- For example, the company requires to obtain 3% more market share with this product, the market segment of the product is small family, the target cost is 5,800 pound, the number of units is 100,000 per year, the target price is 8,000 to 10,000, the potential profit is 2,200 to 4,200 per unit, the product will bring 300,000 to 500,000 customers to the company, the product expected to reduce 1-2% market share of the main competitors, and so on

Figure B.6 – Business Requirements Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Clarified Requirements &Knowledge” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix D and E), when they need to know the contents in the Folder-Based EAF Implementation.

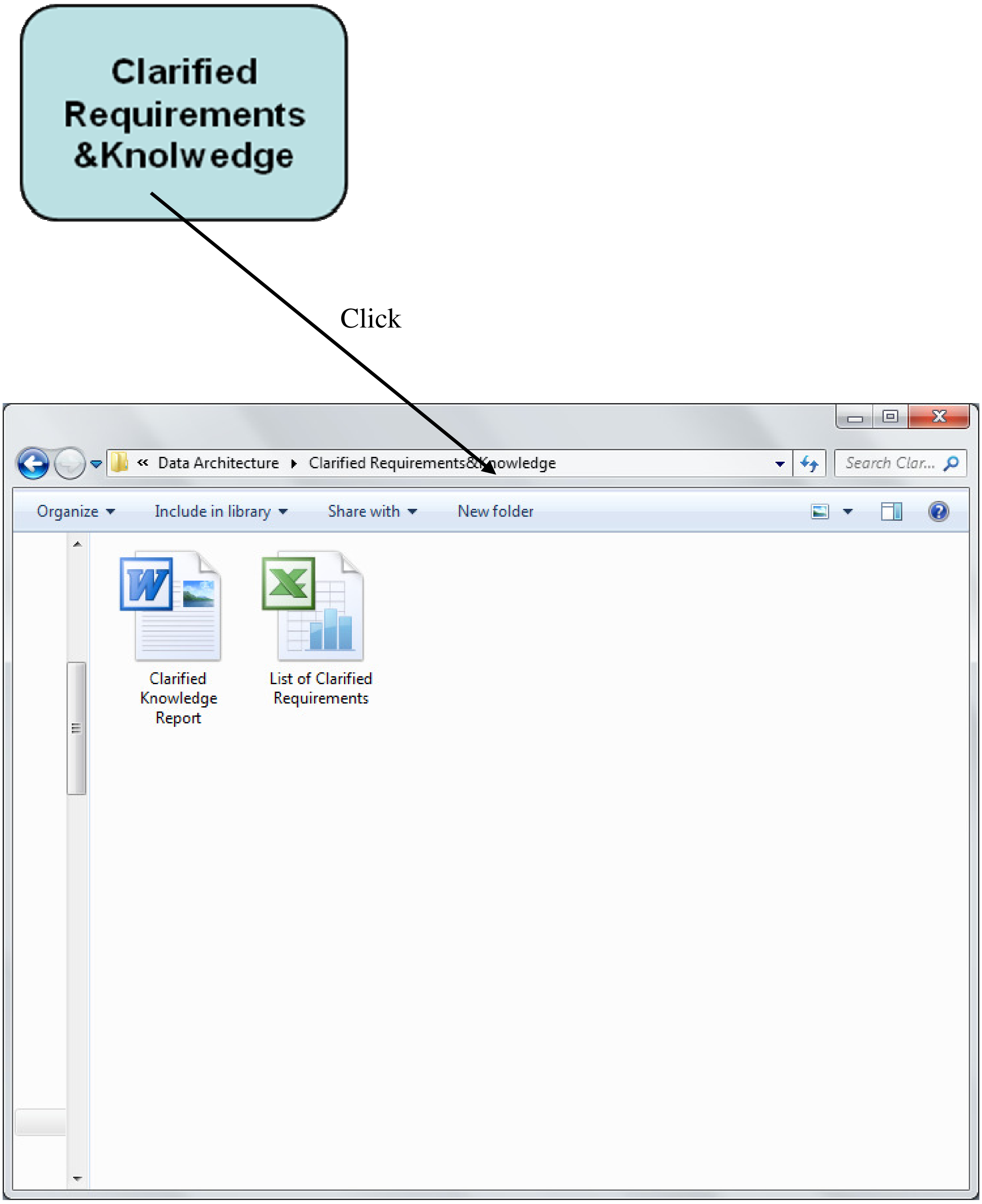

Clarified Requirements &Knolwedge
Click
« Data Architecture ▸ Clarified Requirements&Knowledge
Search Clar...
Organize ▾
Include in library ▾
Share with ▾
New folder
Clarified Knowledge Report
List of Clarified Requirements

Figure B.7 – Clarified Users’ Requirements and Knowledge Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Categorised User’s Requirements & Knowledge” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix D and E), when they need to know the contents in the Folder-Based EAF Implementation.

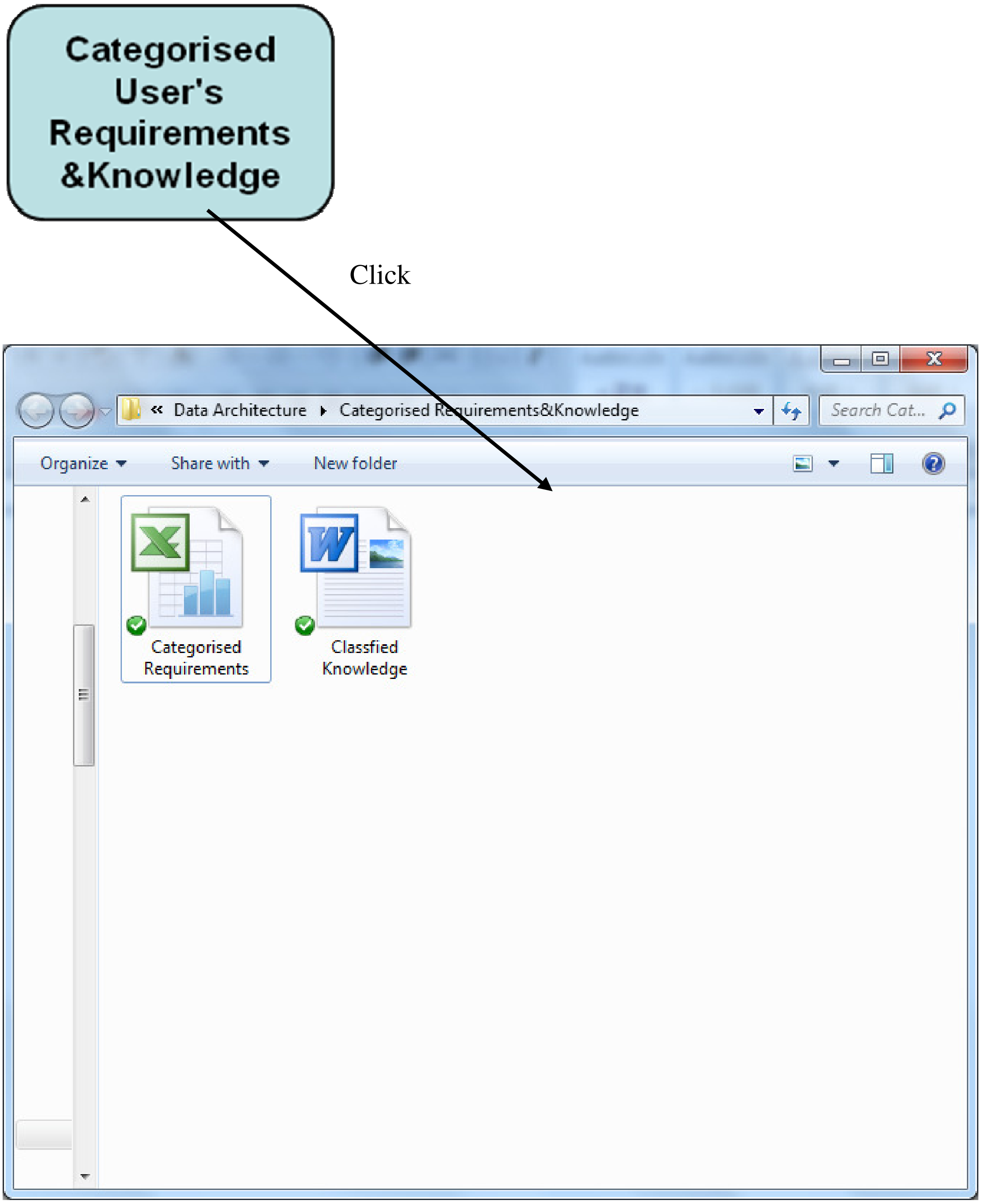

Categorised User's Requirements &Knowledge
Click
« Data Architecture ▸ Categorised Requirements&Knowledge
Search Cat...
Organize ▾
Share with ▾
New folder
Categorised Requirements
Classfied Knowledge

Figure B.8 –Categorised Users’ Requirements and Knowledge Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Integrated Requirements with Knowledge” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix D), when they need to know the contents in the Folder-Based EAF Implementation.

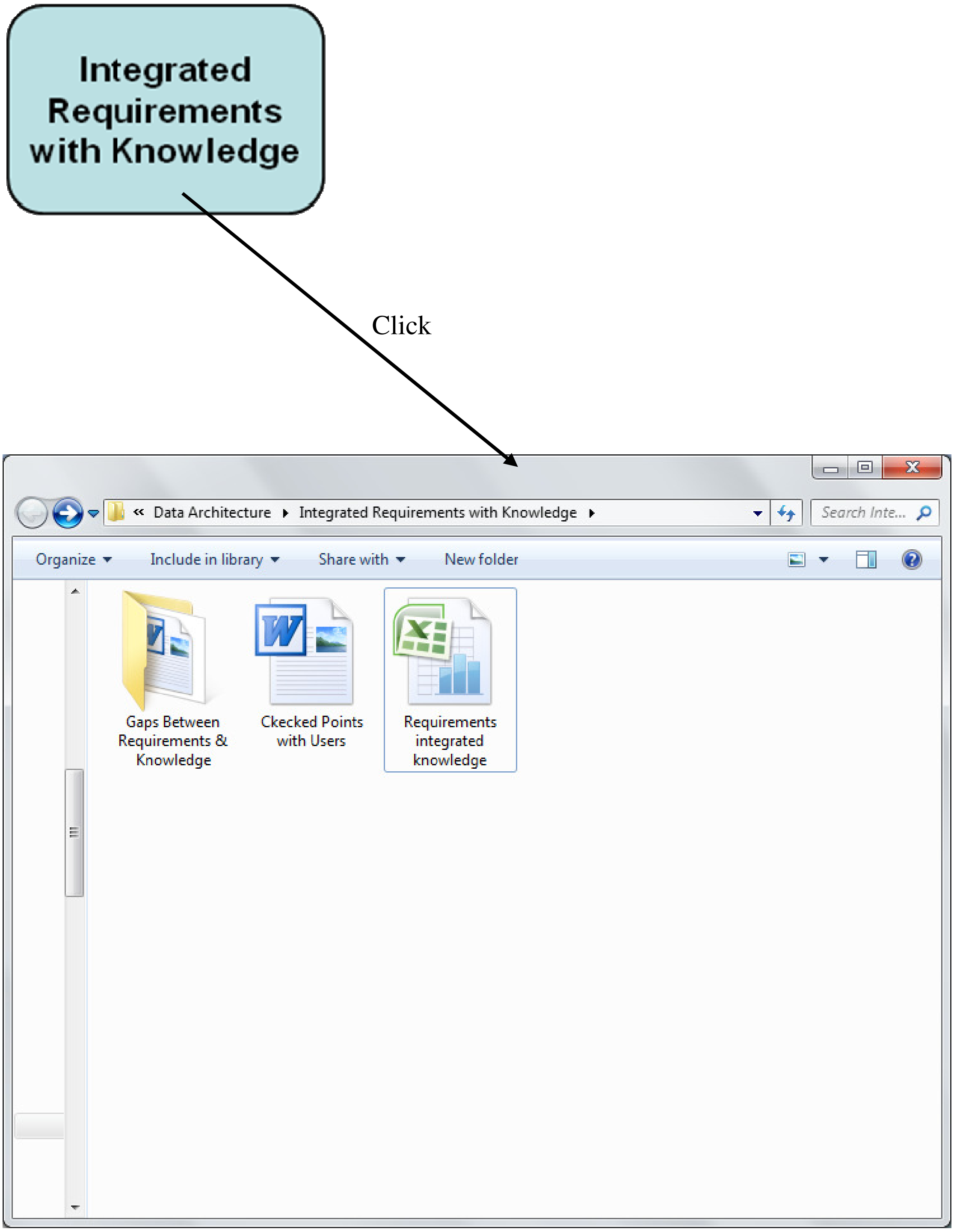

Integrated Requirements with Knowledge
Click
« Data Architecture ▸ Integrated Requirements with Knowledge ▸
Search Inte...
Organize ▾
Include in library ▾
Share with ▾
New folder
Gaps Between Requirements & Knowledge
Ckecked Points with Users
Requirements integrated knowledge

Figure B.9 – Integrated Users’ Requirements with Knowledge Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “List of Prioritised User’s Requirements” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix D), when they need to know the contents in the Folder-Based EAF Implementation.

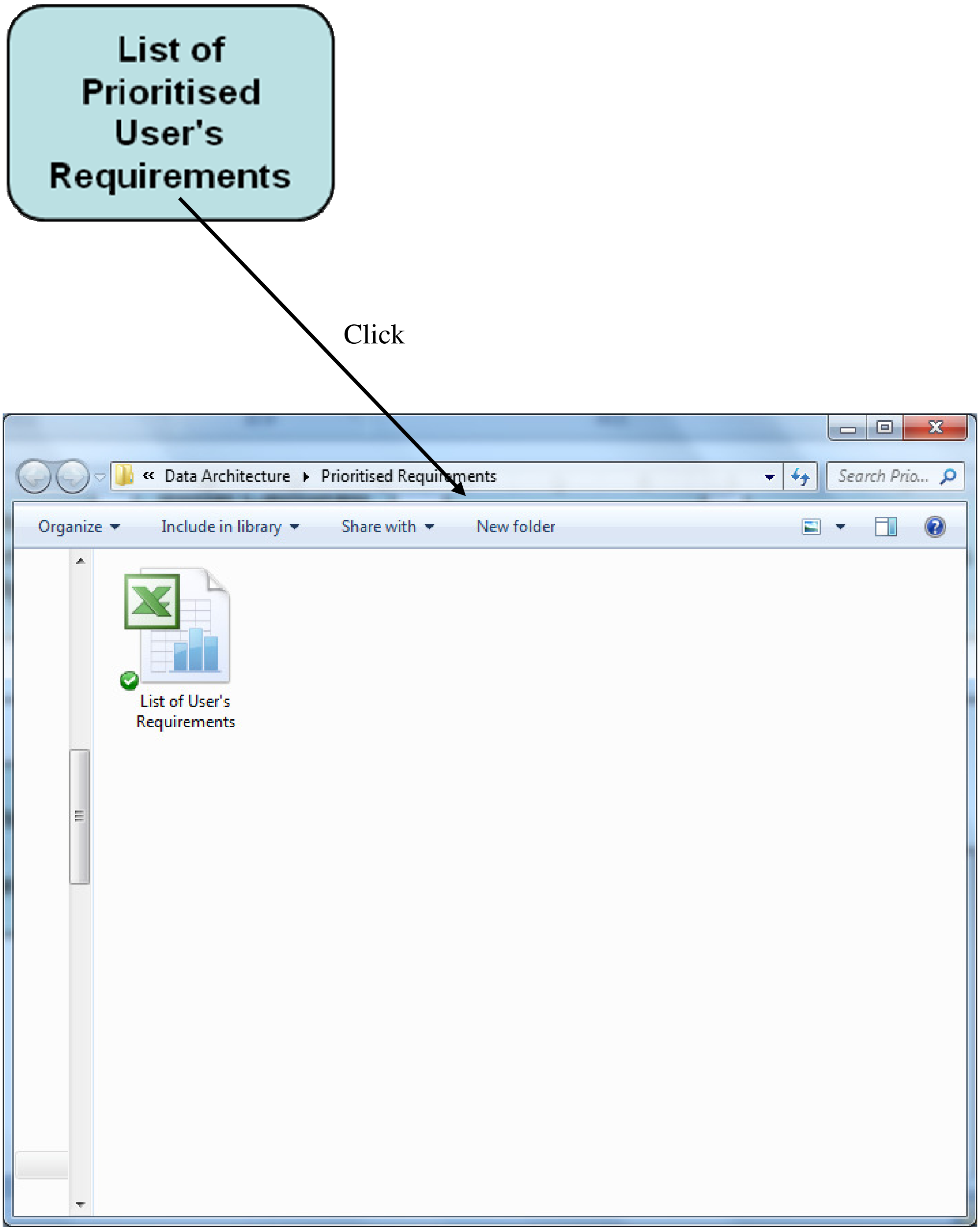

List of Prioritised User's Requirements
Click
« Data Architecture ▸ Prioritised Requirements
Search Prio...
Organize ▾
Include in library ▾
Share with ▾
New folder
List of User's Requirements

Figure B.10 – List of Prioritised Users’ Requirements Implemented in Folder-based Implementation

This screenshot represents that system developers apply the "Final User's Requirements" Feature of the Guideline in the "Folder-based Implementation" with examples (detailed examples are provided in Appendix D), when they need to know the contents in the Folder-Based EAF Implementation.

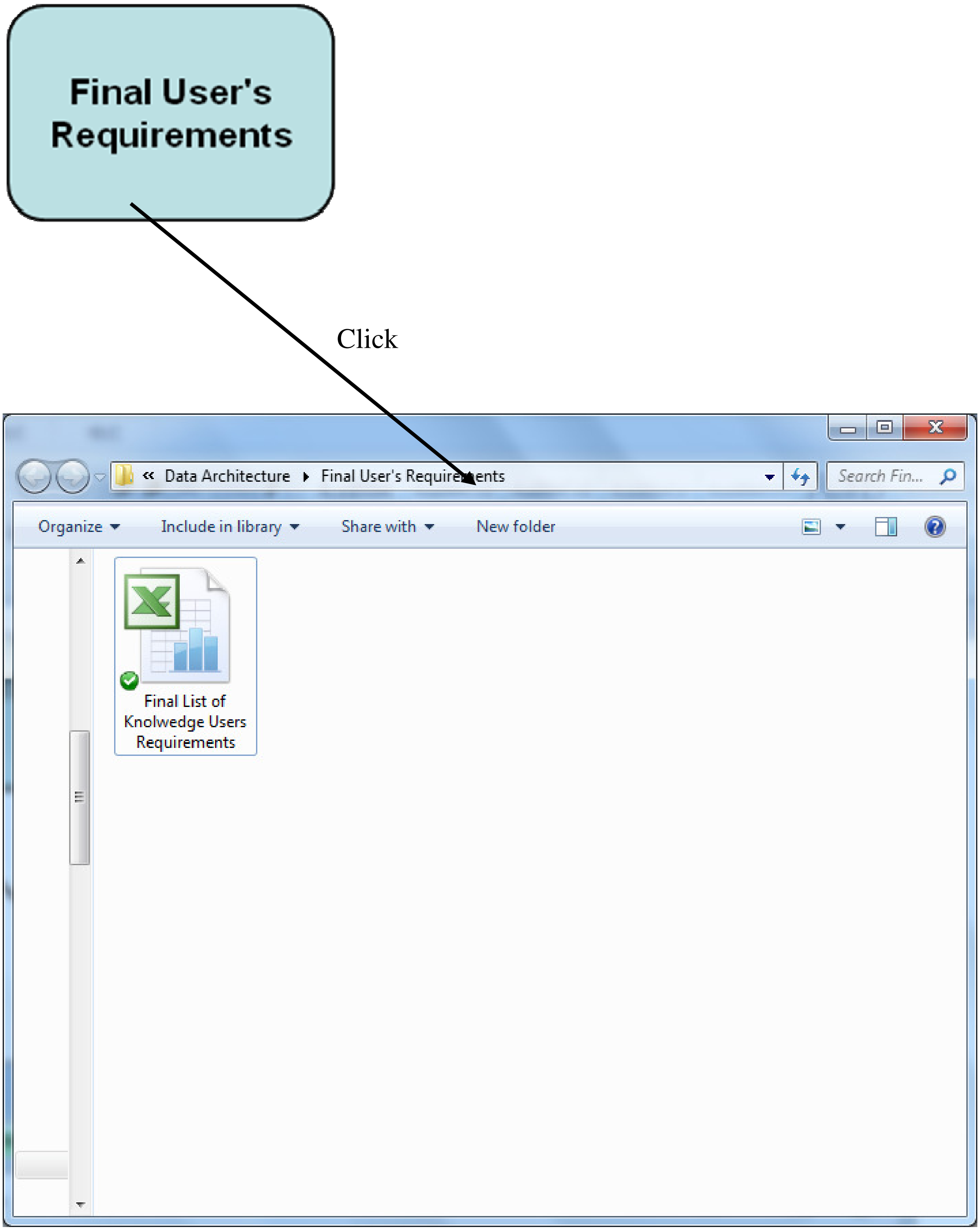

Final User's Requirements
Click
« Data Architecture ▸ Final User's Requirements
Search Fin...
Organize ▾
Include in library ▾
Share with ▾
New folder
Final List of Knolwedge Users Requirements

Figure B.11 – Final List of Users' Requirements Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Functional Requirements” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix F), when they need to know the contents in the Folder-Based EAF Implementation.

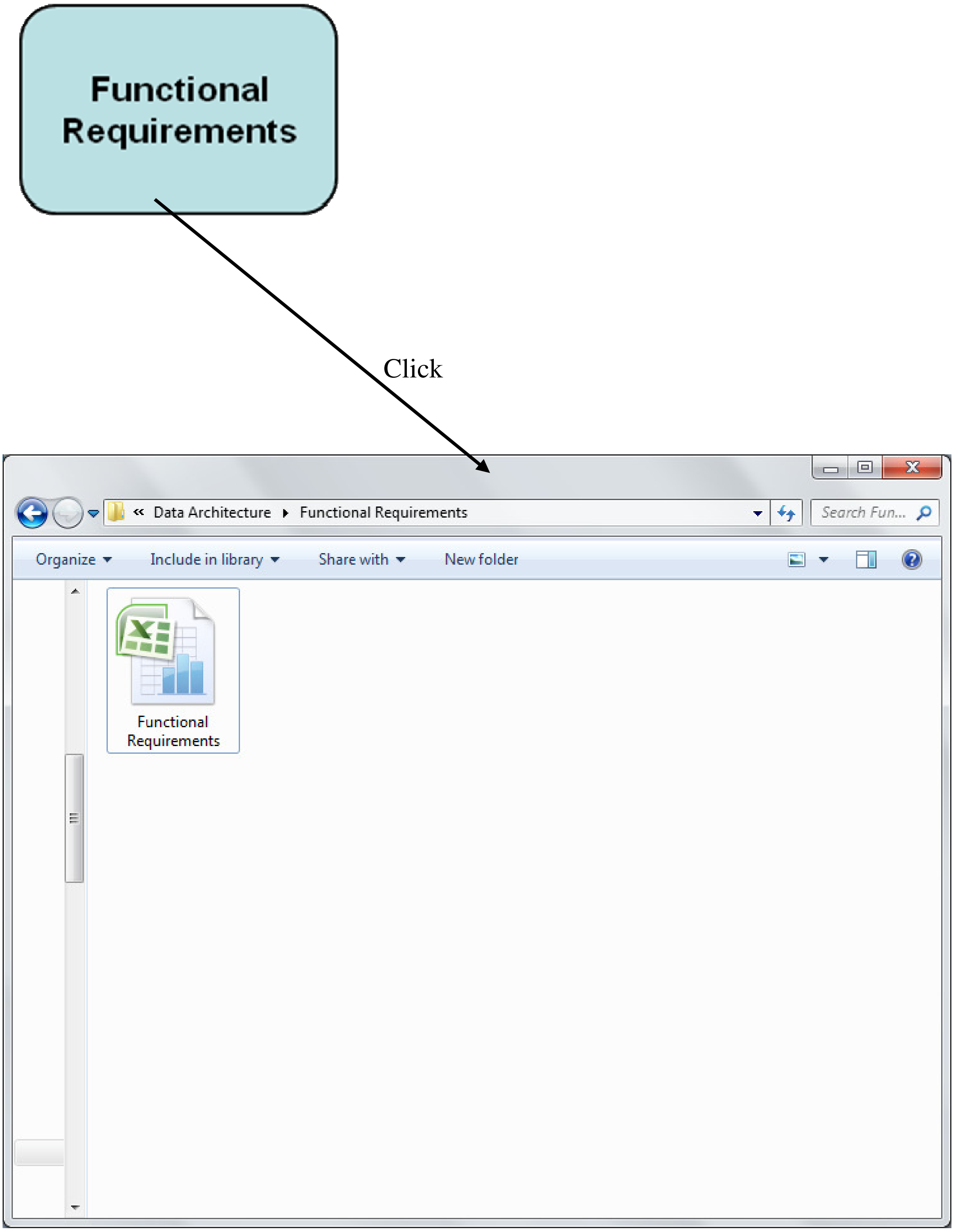


Figure B.12 – Functional Requirements Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Functional Specification” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix F), when they need to know the contents in the Folder-Based EAF Implementation.

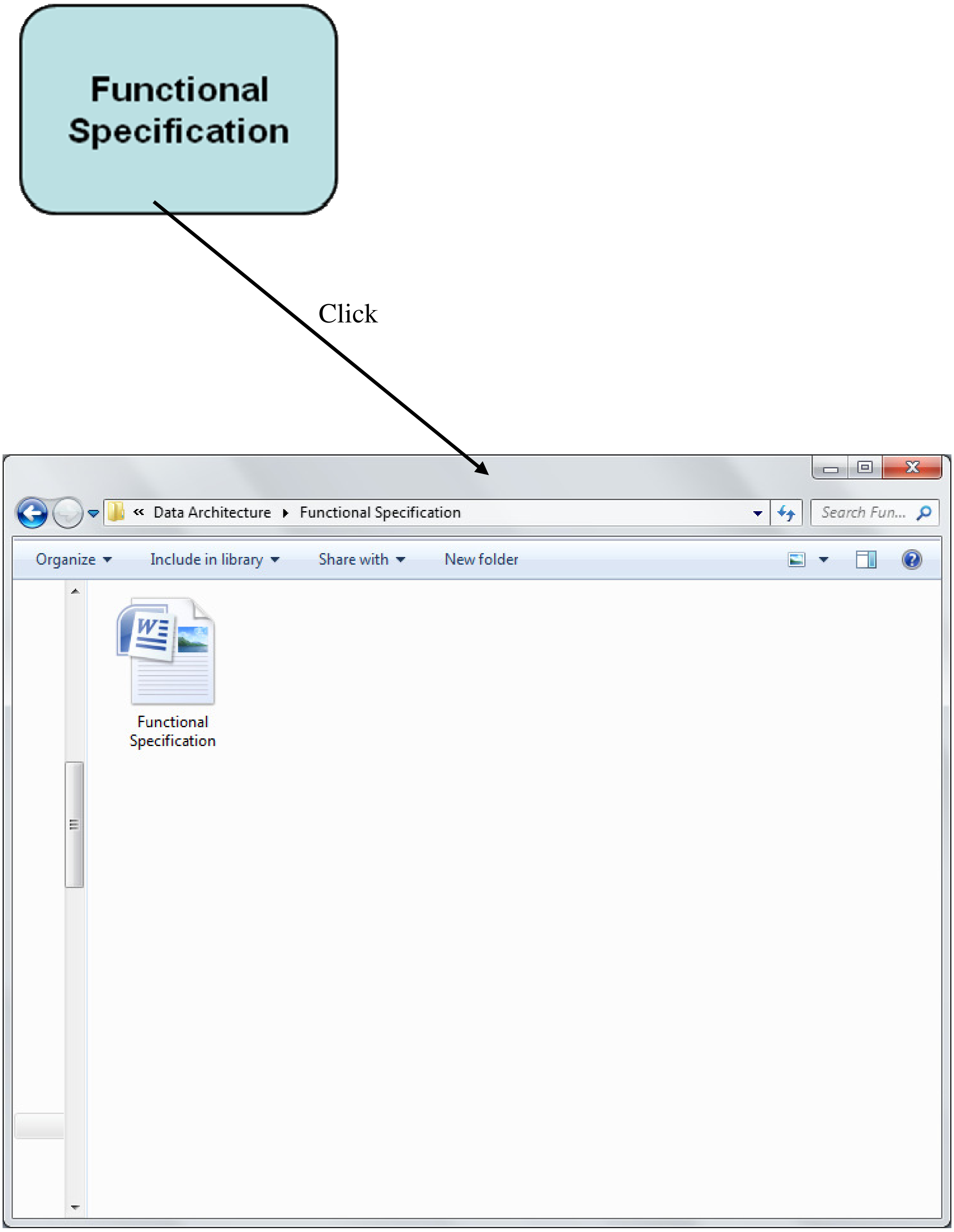


Figure B.13 – Functional Specification Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Improved Functional Specification” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix F), when they need to know the contents in the Folder-Based EAF Implementation.

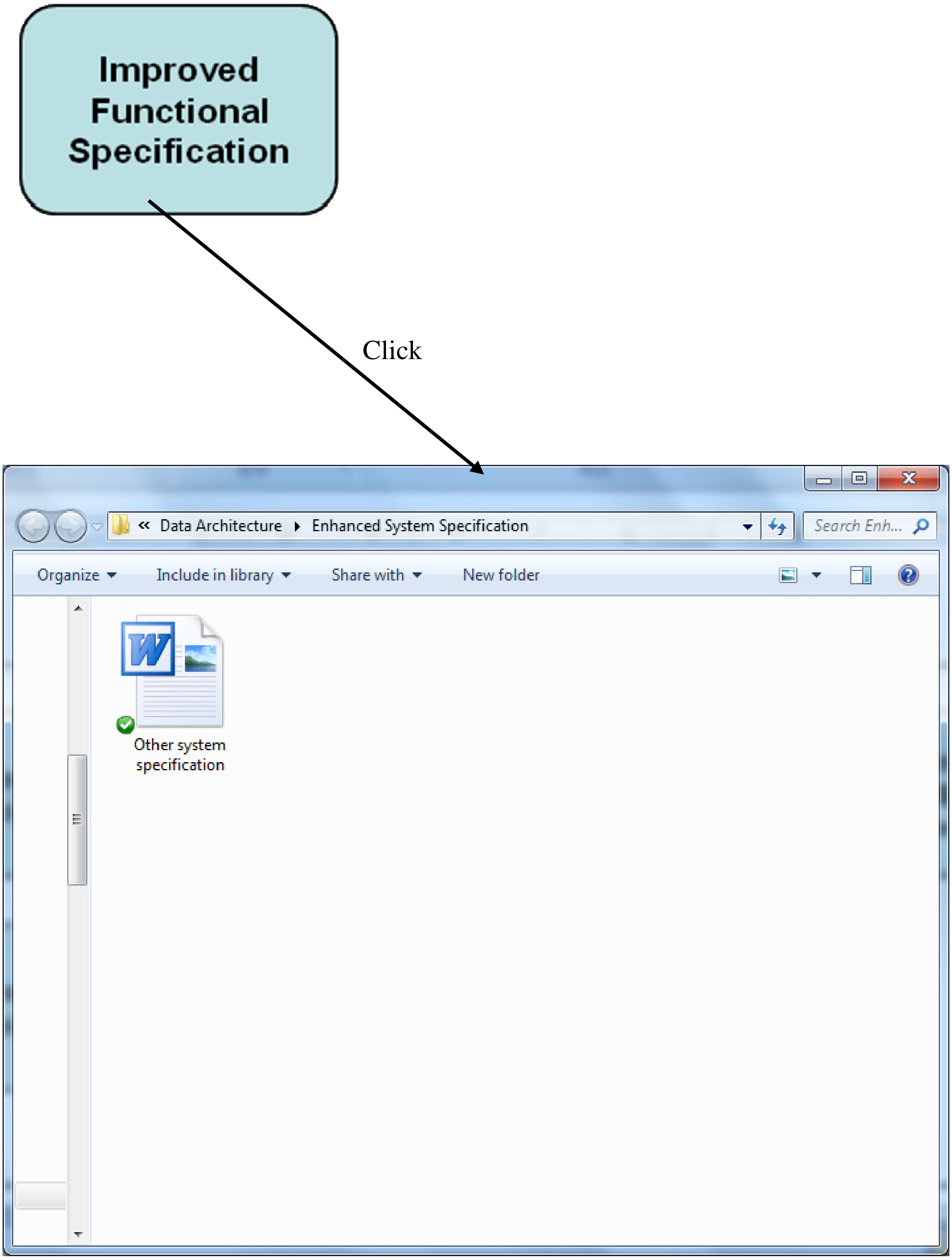

Improved Functional Specification
Click
« Data Architecture ▸ Enhanced System Specification
Search Enh...
Organize
Include in library
Share with
New folder
Other system specification

Figure B.14 – Improved Functional Specification Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “System Configuration” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix F), when they need to know the contents in the Folder-Based EAF Implementation.

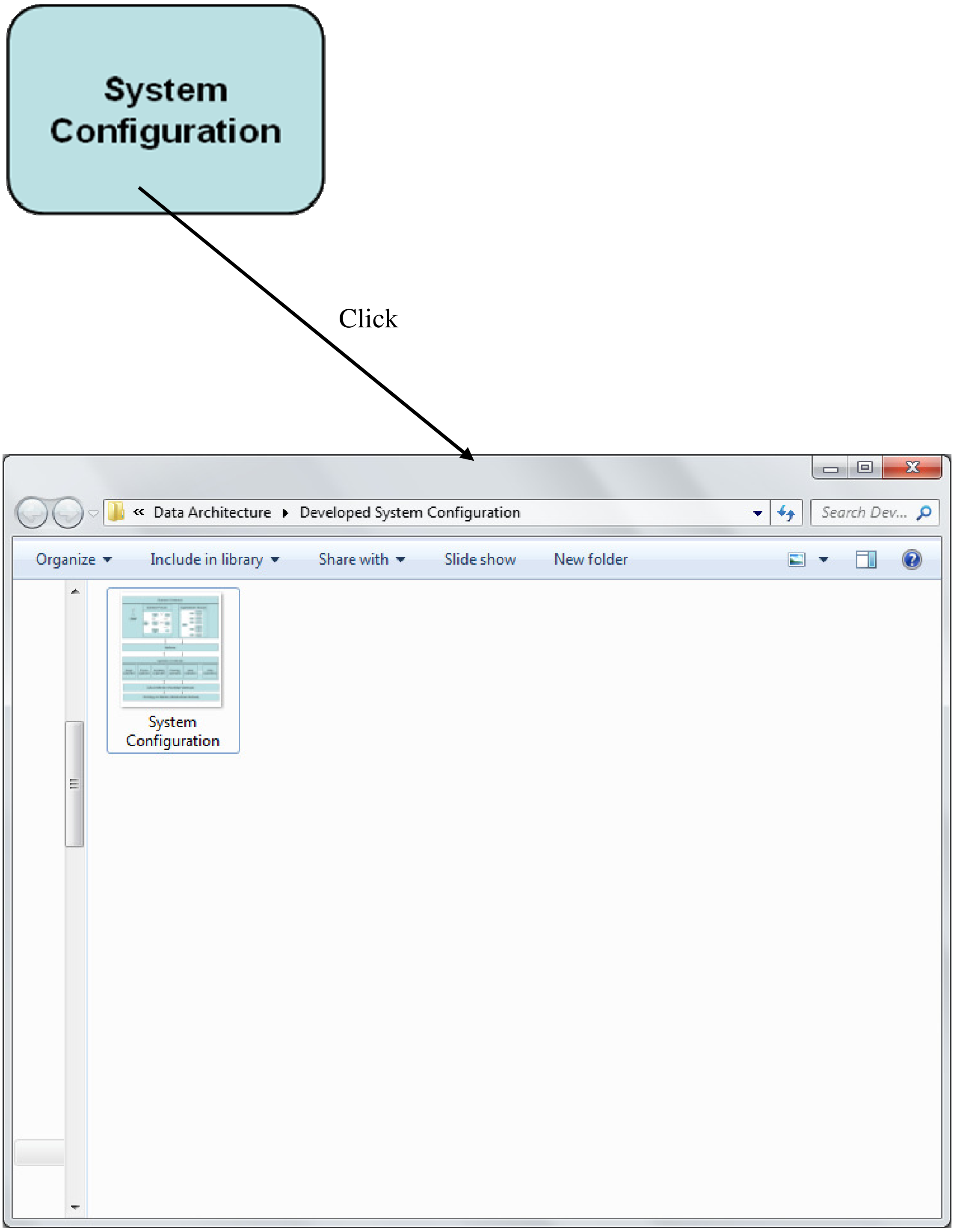


Figure B.15 – System Configuration Implemented in Folder-based Implementation

This screenshot represents that system developers apply the “Detailed System Specification” Feature of the Guideline in the “Folder-based Implementation” with examples (detailed examples are provided in Appendix H), when they need to know the contents in the Folder-Based EAF Implementation.

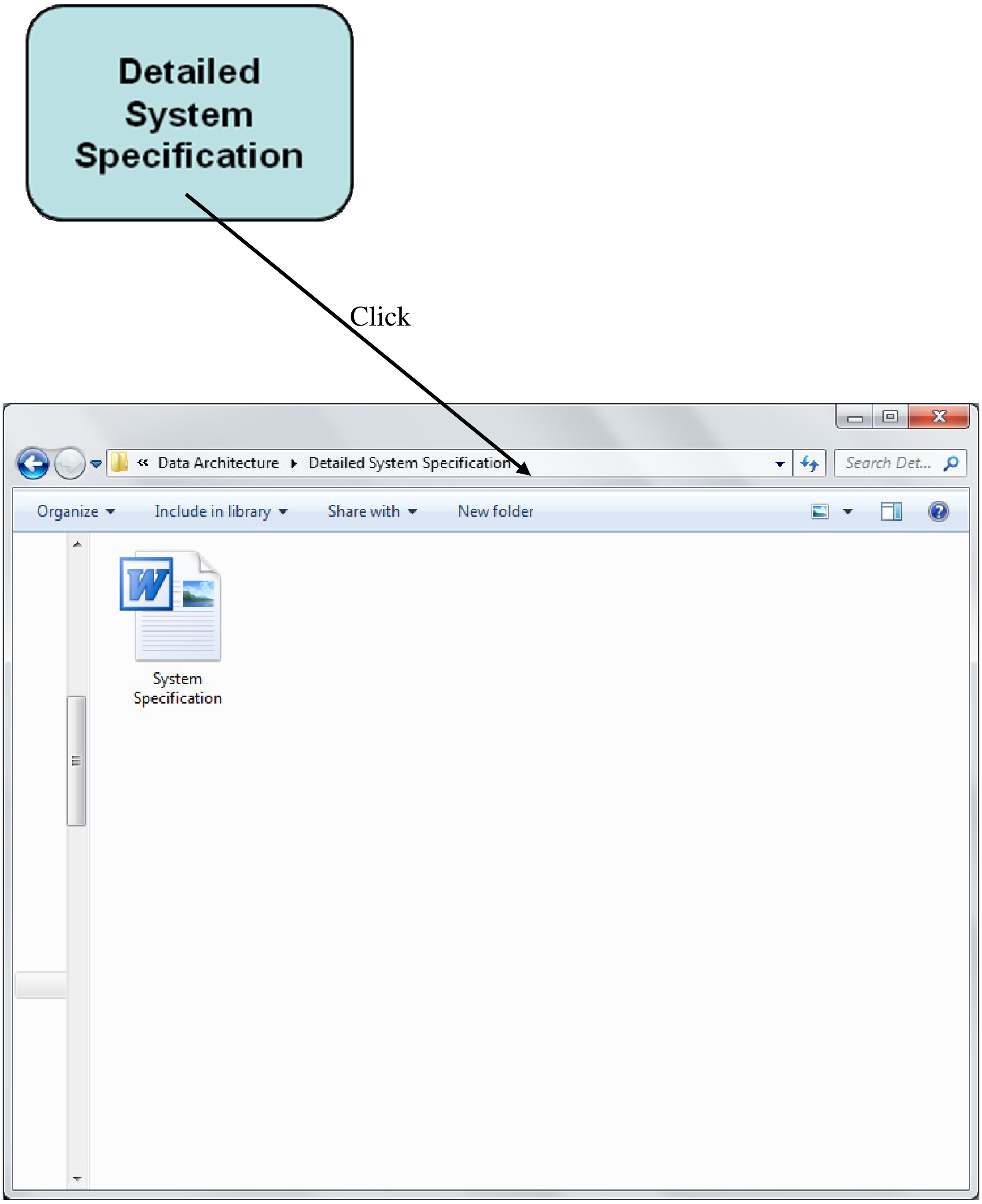
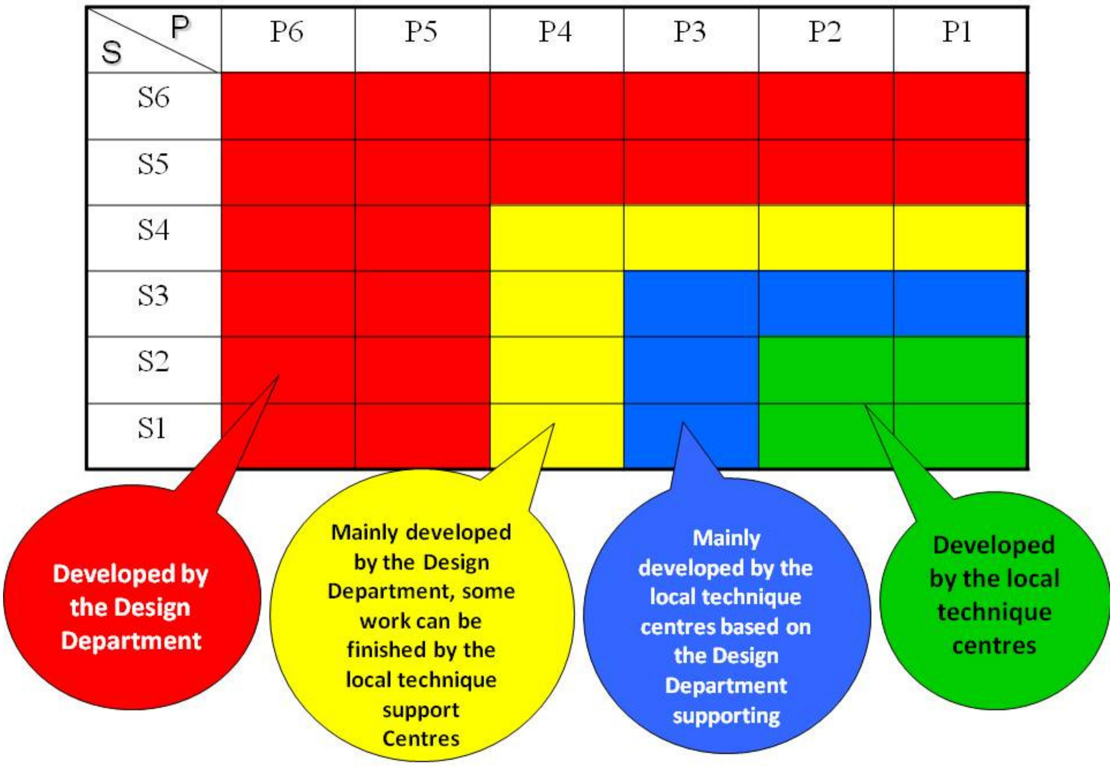


Figure B.16– Detailed System Specification Implemented in Folder-based Implementation

Appendix C: 6 Level Product Development Standardisation in the Collaborating Company



P S	P6	P5	P4	P3	P2	P1
S6	🟥	🟥	🟥	🟥	🟥	🟥
S5	🟥	🟥	🟥	🟥	🟥	🟥
S4	🟥	🟥	🟨	🟨	🟨	🟨
S3	🟥	🟥	🟨	🟦	🟦	🟦
S2	🟥	🟥	🟨	🟦	🟩	🟩
S1	🟥	🟥	🟨	🟦	🟩	🟩

Developed by the Design Department
Mainly developed by the Design Department, some work can be finished by the local technique support Centres
Mainly developed by the local technique centres based on the Design Department supporting
Developed by the local technique centres

Figure C.1 – 6 Levels Standardisation of Product Development in the Collaborating Company

Appendix D: The Full Lists of Knowledge User's Requirements in Each Development Stage

Initially Captured Users' Requirements

Enterprise Level -- Product Manager

Main objective -- to manage and coordinate the whole product development

Initial User's Requirements:

E_P_M1. To model the product lifecycle development process

E_P_M2. To manage the product development process and procedures

E_P_M3. To estimate cost and control budget of the product development project

E_P_M4. To plan human resource based on functions and positions in the process and production line

E_P_M5. To set up check points and milestones in the product development process

E_P_M6. To cooperate and communicate with departments and the board of directors

E_P_M7. To classify functions of departments and allocate tasks to each department and group

E_P_M8. To identify the authorisation of the product knowledge

E_P_M9. To obtain the relevant product knowledge of benchmarking and previous projects

E_P_M10. Product knowledge general management, such as store, search, invoke, integrate and share

Department Level Users' Requirements

Strategic Development Department

Main objective of this department – product analysis

Initial User's Requirements:

S_D_D1. Model product developing process

S_D_D2. Multiple departments collaboration: pending tasks based on prioritisation, constraints set-up based on all of departments

S_D_D3. Functions needs to be included in the interfaces: planning, business process in the department, hierarchical management with functions in the department

S_D_D4. Possible system functions to be needed: synchronisation, notice message, input and output, feedback, and action based on task

S_D_D5. Manage industry research and analyses report

S_D_D6. Gather, manage and integrate macro information and policy

Product Management Department

Main objective – Planning stage especially for concept design and development, planning the process of the product

Initial User's Requirements:

P_M_D1. Benchmark with exiting similar products in the knowledge base including both in-house products and competitor's products

P_M_D2. Knowledge integration required

P_M_D3. Manage knowledge based to store the graphs and figure

P_M_D4. Need a method to search and invoke the relevant knowledge accurately and efficiently

P_M_D5. In-house knowledge store

P_M_D6. Visibility of the resource and controlling

P_M_D7. Digitalise graphs and figures – paperless work

P_M_D8. Require the concept development module in order to create a standard for the concept development

P_M_D9. Need more improvement in usability particularly in user interface and its menus, unclear system structure, input inconvenient, lack communication, hard to search

P_M_D10. OA should be integrated with KM System

Design Department

Main objective – Design for the whole product, includes the Car-Body Centre which is one of the most important groups in the whole company

Initial User's Requirements:

D_D1. Optimise product development process (in an ideal way which is automatic optimisation with system)

D_D2. Simultaneity design and development with other departments and groups

D_D3. Communications with different product developing periods

D_D4. Planning simultaneity integration

D_D5. Multiple departments collaboration especially in product concept design

D_D6. Knowledge base is refreshed and shared rapidly

D_D7. Double checks with message and phone call

D_D8. Paperless work

D_D9. Knowledge Integration – especially for training purpose

D_D10. Modelling process or design in process required

D_D11.System integration – too many small systems in department, almost every department has an individual system

D_D12. Sockets are not match perfectly among sub-systems

Manufacturing department

Main Objective -- Manufacturing Activity

Initial User's Requirements:

MANU_D1. Diagram can be printed as the specific requirements

MANU_D2. Diagram can be double checked with the relevant departments or staffs before allocating

MANU_D3. Workshop leader can input the knowledge summary of the product

MANU_D4. The system can note each change when the product is processing in the product line

MANU_D5. Model manufacturing organisational structure and process

MANU_D6. Plan and Model manufacturing resource and cost about the whole product line

MANU_D7. Manage and plan the raw material and resource included tracking logistics

Sale and Marketing department

Main Objective -- Sale and Marketing Activity

Initial User's Requirements:

S_MAR_D1. Department can search and invoke the necessary knowledge when the department needs to research the market, customers and competitors

S_MAR_D2. Create standard and template for customer interview and feedback, and benchmarking of competitors and potential markets

S_MAR_D3. Integrated all of the relevant knowledge and manage it

S_MAR_D4. The head of department can input the summary of the knowledge to support the product analysis

S_MAR_D5. The head of department can be integrated the product knowledge to existing product knowledge in the system

S_MAR_D6. Manage the researches of markets and customers

Group Level Users' Requirements

Two Groups of the Product Management Department involved as an example

Product Concept Development Group of Product Management Department

Main objective – concept design and development

Initial User's Requirements:

CG_PMD01. Develop a common framework for product concept development as a standard

CG_PMD02. Create a standard for product concept development process

CG_PMD03. Benchmark both in-house and outside product concept based on the system

CG_PMD04. Communication between different departments and groups

CG_PMD05. Require a common platform for better developing and communicating

Product Planning Group of Product Management department

Main objective – Planning stage and planning the process of the product

Initial User's Requirements:

PPG_PMD01. Product manufacturing process planning

PPG_PMD02. Resource planning includes material planning, supplier, and logistics

PPG_PMD03. Labour sources planning and it needs to communicate with HR Department

PPG_PMD04. Cost and Budget planning and communicate with Finance Department

List of Clarified Requirements

Enterprise Level -- Product Manager

Main objective -- to manage and coordinate the whole product development

E_P_M1. To model the product lifecycle development process (95+60 =155)

E_P_M2. To manage the product development process and procedures (85+65 =150)

E_P_M3. To estimate cost and control budget of the product development project (80+55 =145)

E_P_M4. To plan human resource based on functions and positions in the process and production line (80+65 =145)

E_P_M5. To set up check points and milestones in the product development process (70+60 =130)

E_P_M6. To cooperate and communicate with departments and the board of directors (70+50 =120)

E_P_M7. To classify functions of departments and allocate tasks to each department and group (65+60 =125)

E_P_M8. To identify the authorisation of the product knowledge (60+60=120)

E_P_M9. To obtain the relevant product knowledge of benchmarking and previous projects (80+ 45 =125)

E_P_M10. Product knowledge general management, such as store, search, invoke, integrate and share (80+75 =155)

Department Level Users' Requirements

Strategic Development Department

Main objective of this department – product analysis

S_D_D1. Model product developing process (95+65=150)

S_D_D2. Multiple departments collaboration: pending tasks based on prioritisation, constraints set-up based on all of departments (65+50=115)

S_D_D3. Functions needs to be included in the interfaces: planning, business process in the department, hierarchical management with functions in the department (65+50=115)

S_D_D4. Possible system functions to be needed: synchronisation, notice message, input and output, feedback, and action based on task (60+60=120)

S_D_D5. Manage industry research and analyses report (50+55=105)

S_D_D6. Gather, manage and integrate macro information and policy (60+55=115)

Product Management Department

Main objective – Planning stage especially for concept design and development, planning the process of the product

P_M_D1. Benchmark with exiting similar products in the knowledge base including both in-house products and competitor's products (90+45=135)

P_M_D2. Knowledge integration required (75+45=120)

P_M_D3. Manage knowledge based to store the graphs and figure (60+45=105)

P_M_D4. Need a method to search and invoke the relevant knowledge accurately and efficiently (75+40=115)

P_M_D5. In-house knowledge store (80+50=130)

P_M_D6. Visibility of the resource and controlling (50+45=95)

P_M_D7. Digitalise graphs and figures – paperless work (70+45=115)

P_M_D8. Require the concept development module in order to create a standard for the concept development (85+70=155)

P_M_D9. Need more improvement in usability particularly in user interface and its menus, unclear system structure, input inconvenient, lack communication, hard to search (50+45=95)

P_M_D10. OA should be integrated with KM System (50+40=90)

Design Department

Main objective – Design for the whole product, includes the Car-Body Centre which is one of the most important groups in the whole company

D_D1. Optimise product development process (in an ideal way which is automatic optimisation with system) (85+55=140)

D_D2. Simultaneity design and development with other departments and groups (75+35=110)

D_D3. Communications with different product developing periods (50+40=90)

D_D4. Planning simultaneity integration (60+35=95)

D_D5. Multiple departments collaboration especially in product concept design (60+40=100)

D_D6. Knowledge base is refreshed and shared rapidly (50+45=95)

D_D7. Double checks with message and phone call (40+45=85)

D_D8. Paperless work (55+45=100)

D_D9. Knowledge Integration – especially for training purpose (75+35=110)

D_D10. Modelling process or design in process required (85+65=150)

D_D11.System integration – too many small systems in department, almost every department has an individual system (40+30=70)

D_D12. Sockets are not match perfectly among sub-systems (35+40=75)

Manufacturing department

Main Objective -- Manufacturing Activity

MANU_D1. Diagram can be printed as the specific requirements (60+35=95)

MANU_D2. Diagram can be double checked with the relevant departments or staffs before allocating (60+55=115)

MANU_D3. Workshop leader can input the knowledge summary of the product (60+60=120)

MANU_D4. The system can note each change when the product is processing in the product line (75+50=125)

MANU_D5. Model manufacturing organisational structure and process (80+65=145)

MANU_D6. Plan and Model manufacturing resource and cost about the whole production line (70+45=115)

MANU_D7. Manage and plan the raw material and resource included tracking logistics (75+45=120)

Sale and Marketing department

Main Objective -- Sale and Marketing Activity

S_MAR_D1. Department can search and invoke the necessary knowledge when the department needs to research the market, customers and competitors (80+60=130)

S_MAR_D2. Create standard and template for customer interview and feedback, and benchmarking of competitors and potential markets (85+60=145)

S_MAR_D3. Integrated all of the relevant knowledge and manage it (65+45=110)

S_MAR_D4. The head of department can input the summary of the knowledge to support the product analysis (65+50=115)

S_MAR_D5. The head of department can be integrated the product knowledge to existing product knowledge in the system (65+45=110)

S_MAR_D6. Manage the researches of markets and customers (65+60=125)

Group Level Users' Requirements

Two Groups of the Product Management Department involved as an example

Product Concept Development Group of Product Management Department

Main objective – concept design and development

CG_PMD01. Develop a common framework for product concept development as a standard (95+50=145)

CG_PMD02. Create a standard for product concept development process (60+45=105)

CG_PMD03. Benchmark both in-house and outside product concept based on the system (75+35=110)

CG_PMD04. Communication between different departments and groups (55+55=110)

CG_PMD05. Require a common platform for better developing and communicating (55/50=105)

Product Planning Group of Product Management department

Main objective – Planning stage and planning the process of the product

PPG_PMD01. Product manufacturing process planning (95+60=155)

PPG_PMD02. Resource planning includes material planning, supplier, and logistics (75+65=140)

PPG_PMD03. Labour sources planning and it needs to communicate with HR Department (65+40=105)

PPG_PMD04. Cost and Budget planning and communicate with Finance Department (70+45=115)

Categorised Requirements

Initial Requirement ID	Content	Relevant Department /Group	Importance Point
Enterprise Level			
E_P_M1	To Model and manage the developing product lifecycle	Enterprise	155
E_P_M2/PPG_PMD01	To manage the product Development process and procedure	Enterprise/PMD	155 (155,150)
E_P_M3/PPG_PMD04	To estimate cost and Control budget of the product development project	Enterprise/PMD	145 (145,115)
E_P_M4/MANU_D6/PPG_PMD03	To plan and manage human resource based on functions and positions in the process and product line	Enterprise/ Manufacturing	145(145,115,105)
E_P_M5	To set up check points and milestones in the product development process	Enterprise	130
E_P_M6/S_D_D2/CG_PMD04	To cooperate and communicate with departments and the chair board vertically and horizontally	Enterprise/Strategic Development Department/ PMD	120(120,115,110)
E_P_M7	To classify functions of departments and allocate tasks of each department and group	Enterprise	125
E_P_M8	To identify the authorisation of the product knowledge	Enterprise	120

E_P_M9	To obtain the relevant product knowledge of benchmarking and previous projects	Enterprise	125
E_P_M10/P_M_D2/P_M_D4/P_M_D5/D_D9/MANU_D3/S_MAR_D3/S_MAR_D4/S_MAR_D5	Product knowledge general management, such as store, search, invoke, integrate, share and etc	Enterprise(Product Manager)/Product Management Department/ Design Department/ Manufacturing Department/ Sale and Marketing Department	155 (155,120,115,130,110,120,110,115,110)
Department Level			
S_D_D1	Simulating the product developing process	Strategic Development Department	150
S_D_D3.1	Functions needs to be included in the interfaces: planning business process in the department,	Strategic Development Department	125
S_D_D3.2	hierarchical management with functions in the department	Strategic Development Department	115
S_D_D4	Possible system functions to be needed: synchronisation, notice message, input and output, feedback, and action based on task	Strategic Development Department	120
S_D_D5	To Manage industry research and analysis report	Strategic Development Department	105
S_D_D6	To gather, manage and integrate macro information and policy	Strategic Development Department	115

P_M_D1	benchmarking – with the similar product in the knowledge base including both in-house product and competitor's product	Product Management Department	135
P_M_D3	large knowledge based to store the graphs and figure	Product Management Department	105
P_M_D6	visibility of the resource and controlling	Product Management Department	95
P_M_D7/D_D8	digitalise graphs and figures – paperless work	Product Management Department	115 (115,100)
P_M_D8	Require the concept development module in order to create a standard for the concept development	Product Management Department	155
P_M_D9	Needs more improvement in usability particularly in user interface and its menus, Unclear system structure, input inconvenient, lack communication, hard to search	Product Management Department	95
P_M_D10	OA should be integrated with KM System	Product Management Department	90
D_D1	optimising product development process, the ideal way is automatic optimisation with system	Design Department	140
D_D2	simultaneity design and development with other departments and groups	Design Department	110
D_D3	communication and logical link with different product developing periods	Design Department	90

D_D4	Planning simultaneity integration	Design Department	95
D_D5	multiple departments collaboration especially in product concept design	Design Department	100
D_D6	Knowledge base is refreshed and shared rapidly	Design Department	95
D_D7	Double checks with message and phone call	Design Department	85
D_D10	Simulating process or design in process required	Design Department	150
D_D11	System integration – too many small systems in department, almost every departments has an individual system	Design Department	70
D_D12	Sockets are not match perfectly among sub-systems	Design Department	75
MANU_D1	Diagram can be printed as the specific requirements	Manufacturing Department	95
MANU_D2	Diagram can be double checked with the relevant departments or staffs before allocating	Manufacturing Department	115
MANU_D4	The system can note each changes when the product is processing in the product line	Manufacturing Department	125
MANU_D5	To simulate Manufacturing organisational structure and process	Manufacturing Department	145
MANU_D7	To manage and plan the raw material and resource included tracking logistics	Manufacturing Department	120

S_MAR_D1	The department can search and invoke the necessary knowledge when the department needs to research the market, customers and competitors	Sale and Marketing Department	130
S_MAR_D2	to create standard and template for customer interview and feedback, and benchmarking of competitors and potential markets	Sale and Marketing Department	145
S_MAR_D6	To manage the researches of markets and customers	Sale and Marketing Department	125
Group Level			
CG_PMD01	Develop a common framework for product concept development as a standard	Product Concept Development Group	145
CG_PMD02	Create a standard for product concept development process	Product Concept Development Group	105
CG_PMD03	Benchmarking both in-house and outside product concept based on the system	Product Concept Development Group	110
CG_PMD05	Require a common platform for better developing and communicating	Product Concept Development Group	105
PPG_PMD02	Resource planning includes material planning, supplier, logistics	Product Planning Group	140

Integrated Requirements with Knowledge

Initial Requirement ID	Content	Enhanced based on Gaps	Relevant Department/Group	Importance Point
Enterprise Level				
E_P_M1	To Model and manage the developing product lifecycle	Not Needed	Enterprise	155
E_P_M2 /PPG_PMD01	To manage the product Development process and procedure	the problems encountered during development process and the methods used to fix the problems	Enterprise &PMD	155 (155,150)
E_P_M3 /PPG_PMD04	To estimate cost and Control budget of the product development project	Not Needed	Enterprise/ PMD	145 (145,115)
E_P_M4 /MANU_D6/PPG_PMD03	To plan and manage human resource based on functions and positions in the process and product line	Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners	Enterprise/ Manufacturing	145(145,115,105)
E_P_M5	To set up check points and milestones in the product development process	In-house product reviews and achievements, some specific knowledge about what this product can bring and why	Enterprise	130
E_P_M6 /S_D_D2/CG_PMD04	To cooperate and communicate with departments and the chair board vertically and horizontally	Not Needed	Enterprise/ Strategic Development Department/PMD	120(120,115,110)

E_P_M7	To classify functions of departments and allocate tasks of each department and group	Not Needed	Enterprise	125
E_P_M8	To identify the authorisation of the product knowledge	Not Needed	Enterprise	120
E_P_M9	To obtain the relevant product knowledge of benchmarking and previous projects	Not Needed	Enterprise	125
E_P_M10/P_M_D2/P_M_D4/P_M_D5/D_D9/MANU_D3/S_MAR_D3/S_MAR_D4/S_MAR_D5	Product knowledge general management, such as store, search, invoke, integrate, share and etc	Specially focus on tacit knowledge: Lessons learned, personal experiences. Some Necessary Explicit knowledge: Standardisations, principles and rules and policies	Enterprise(Product Manager)/ Product Management Department/ Design Department/ Manufacturing Department/Sale and Marketing Department	155 (155,120,115, 130,110,120,110,115,110)
New1	To ensure the potential benefit indication about the new product	New	Enterprise	85+70=155
New2	To manage meeting Munities of each level	New	Enterprise	80+70=150
New3	To manage the Board decisions which are based on the existing in-house product knowledge	New	Enterprise	70+70=140

New4	To simulate the main functional implementation and application of each department	New	Enterprise	85+55=140
Department Level				
S_D_D1	Simulating the product developing process	Not Needed	Strategic Development Department	150
S_D_D3.1	Functions needs to be included in the interfaces: planning business process in the department,	Not Needed	Strategic Development Department	125
S_D_D3.2	hierarchical management with functions in the department	Not Needed	Strategic Development Department	115
S_D_D4	Possible system functions to be needed: synchronisation, notice message, input and output, feedback, and action based on task	Not Needed	Strategic Development Department	120
S_D_D5	To Manage industry research and analysis report	Not Needed	Strategic Development Department	105
S_D_D6	To gather, manage and integrate macro information and policy	Not Needed	Strategic Development Department	115

P_M_D1	benchmarking – with the similar product in the knowledge base including both in-house product and competitor’s product	Not Needed	Product Management Department	135
P_M_D3	large knowledge based to store the graphs and figure	Not Needed	Product Management Department	105
P_M_D6	visibility of the resource and controlling	Not Needed	Product Management Department	95
P_M_D7/D_D8	digitalise graphs and figures – paperless work	Not Needed	Product Management Department	115 (115,100)
P_M_D8	Require the concept development module in order to create a standard for the concept development	Not Needed	Product Management Department	155
P_M_D9	Needs more improvement in usability particularly in user interface and its menus, Unclear system structure, input inconvenient, lack communication, hard to search	Not Needed	Product Management Department	95
P_M_D10	OA should be integrated with KM System	Not Needed	Product Management Department	90

D_D1	optimising product development process, the ideal way is automatic optimisation with system	Not Needed	Design Department	140
D_D2	simultaneity design and development with other departments and groups	Not Needed	Design Department	110
D_D3	communication and logical link with different product developing periods	Not Needed	Design Department	90
D_D4	Planning simultaneity integration	Not Needed	Design Department	95
D_D5	multiple departments collaboration especially in product concept design	Not Needed	Design Department	100
D_D6	Knowledge base is refreshed and shared rapidly	Not Needed	Design Department	95
D_D7	Double checks with message and phone call	Not Needed	Design Department	85
D_D10	Simulating process or design in process required	Not Needed	Design Department	150
D_D11	System integration – too many small systems in department, almost every departments has an individual system	Not Needed	Design Department	70
D_D12	Sockets are not match perfectly among sub-systems	Not Needed	Design Department	75
MANU_D1	Diagram can be printed as the specific requirements	Not Needed	Manufacturing Department	95

MANU_D2	Diagram can be double checked with the relevant departments or staffs before allocating	Not Needed	Manufacturing Department	115
MANU_D4	The system can note each changes when the product is processing in the product line	Not Needed	Manufacturing Department	125
MANU_D5	To simulate Manufacturing organisational structure and process	Not Needed	Manufacturing Department	145
MANU_D7	To manage and plan the raw material and resource included tracking logistics	Not Needed	Manufacturing Department	120
S_MAR_D1	The department can search and invoke the necessary knowledge when the department needs to research the market, customers and competitors	Not Needed	Sale and Marketing Department	130
S_MAR_D2	to create standard and template for customer interview and feedback, and benchmarking of competitors and potential markets	Not Needed	Sale and Marketing Department	145
S_MAR_D6	To manage the researches of markets and customers	Not Needed	Sale and Marketing Department	125
Group Level				

CG_PMD01	Develop a common framework for product concept development as a standard	Not Needed	Product Concept Development Group	145
CG_PMD02	Create a standard for product concept development process	Not Needed	Product Concept Development Group	105
CG_PMD03	Benchmarking both in-house and outside product concept based on the system	Not Needed	Product Concept Development Group	110
CG_PMD05	Require a common platform for better developing and communicating	Not Needed	Product Concept Development Group	105
PPG_PMD02	Resource planning includes material planning, supplier, logistics	Not Needed	Product Planning Group	140

List of Final Prioritised Requirements

User's Requirement ID	Content	Enhanced based on Gaps	Relevant Department/Group	Importance Point	Category (NE/ US/ WT)	Checked
Enterprise Level						
UR01	Product knowledge general management, such as store, search, invoke, integrate, share and etc	Specially focus on tacit knowledge: Lessons learned, personal experiences. Some Necessary Explicit knowledge: Standardisations, principles and rules and policies	Enterprise (Product Manager)/ Product Management Department/ Design Department/ Manufacturing Department/Sale and Marketing Department	155 (155,120,115,130,110,120,110,115,110)	NE	√
UR02	To manage the product Development process and procedure	the problems encountered during development process and the methods used to fix the problems	Enterprise (Product Manager)/ PMD	155 (155,150)	NE	√
UR03	To Model and manage the developing product lifecycle	Not Needed	Enterprise	155	NE	√

UR04	To ensure the potential benefit indication about the new product	New	Enterprise	155	NE	√
UR05	To manage meeting Munities of each level	New	Enterprise	150	NE	√
UR06	To plan and manage human resource based on functions and positions in the process and product line	Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners	Enterprise/ Strategic Development Department/PMD	145(145,115,105)	NE	√
UR07	To estimate cost and Control budget of the product development project	Not Needed	Enterprise/ Manufacturing	145 (145,115)	NE	√
UR08	To manage the Board decisions which are based on the existing in-house product knowledge	New	Enterprise	140	NE	√
UR09	To simulate the main functional implementation and application of each department	New	Enterprise	140	NE	√
UR10	To set up check points and milestones in the product development process	In-house product reviews and achievements, some specific knowledge about what this product can bring and why	Enterprise	130	NE	√

UR11	To classify functions of departments and allocate tasks of each department and group	Not Needed	Enterprise	125	NE	√
UR12	To obtain the relevant product knowledge of benchmarking and previous projects	Not Needed	Enterprise	125	NE	√
UR13	To cooperate and communicate with departments and the chair board vertically and horizontally	Not Needed	Enterprise/ Strategic Development Department/PMD	120(120,115,110)	NE	√
UR14	To identify the authorisation of the product knowledge	Not Needed	Enterprise	120	NE	√
Department Level						
UR15	Require the concept development module in order to create a standard for the concept development	Not Needed	Product Management Department	155	NE	√
UR16	Simulating the product developing process	Not Needed	Strategic Development Department	150	NE	√
UR17	Simulating process or design in process required	Not Needed	Design Department	150	NE	√

UR18	to create standard and template for customer interview and feedback, and benchmarking of competitors and potential markets	Not Needed	Sale and Marketing Department	145	NE	√
UR19	To simulate Manufacturing organisational structure and process	Not Needed	Manufacturing Department	145	NE	√
UR20	optimising product development process, the ideal way is automatic optimisation with system	Not Needed	Design Department	140	NE	√
UR21	benchmarking – with the similar product in the knowledge base including both in-house product and competitor's product	Not Needed	Product Management Department	135	NE	√
UR22	The department can search and invoke the necessary knowledge when the department needs to research the market, customers and competitors	Not Needed	Sale and Marketing Department	130	NE	√
UR23	To manage the researches of markets and customers	Not Needed	Sale and Marketing Department	125	NE	√

UR24	The system can note each changes when the product is processing in the product line	Not Needed	Manufacturing Department	125	NE	√
UR25	To manage and plan the raw material and resource included tracking logistics	Not Needed	Manufacturing Department	120	NE	√
UR26	digitalise graphs and figures – paperless work	Not Needed	Product Management Department	115(115,100)	US	√
UR27	hierarchical management with functions in the department	Not Needed	Strategic Development Department	115	US	√
UR28	To gather, manage and integrate macro information and policy	Not Needed	Strategic Development Department	115	US	√
UR29	Diagram can be double checked with the relevant departments or staffs before allocating	Not Needed	Manufacturing Department	115	US	√
UR30	simultaneity design and development with other departments and groups	Not Needed	Design Department	110	US	√
UR31	To Manage industry research and analysis report	Not Needed	Strategic Development Department	105	US	√

UR32	large knowledge based to store the graphs and figure	Not Needed	Product Management Department	105	US	√
UR33	multiple departments collaboration especially in product concept design	Not Needed	Design Department	100	US	√
UR34	visibility of the resource and controlling	Not Needed	Product Management Department	95	WT	√
UR35	Needs more improvement in usability particularly in user interface and its menus, Unclear system structure, input inconvenient, lack communication, hard to search	Not Needed	Product Management Department	95	WT	√
UR36	Planning simultaneity integration	Not Needed	Design Department	95	WT	√
UR37	Diagram can be printed as the specific requirements	Not Needed	Manufacturing Department	95	WT	√
UR38	Knowledge base is refreshed and shared rapidly	Not Needed	Design Department	95	WT	√

UR39	OA should be integrated with KM System	Not Needed	Product Management Department	90	WT	√
UR40	communication and logical link with different product developing periods	Not Needed	Design Department	90	WT	√
UR41	Double checks with message and phone call	Not Needed	Design Department	85	WT	√
UR42	Sockets are not match perfectly among sub-systems	Not Needed	Design Department	75	WT	√
UR43	System integration – too many small systems in department, almost every departments has an individual system	Not Needed	Design Department	70	WT	√
Group Level						
UR44	Develop a common framework for product concept development as a standard	Not Needed	Product Concept Development Group of Product Management Department	145	NE	√
UR45	Resource planning includes material planning, supplier, logistics	Not Needed	Product Planning Group	140	NE	√

UR46	Benchmarking both in-house and outside product concept based on the system	Not Needed	Product Concept Developm ent Group of Product Managem ent Departme nt	110	US	√
UR47	Create a standard for product concept development process	Not Needed	Product Concept Developm ent Group of Product Managem ent Departme nt	105	US	√
UR48	Require a common platform for better developing and communicating	Not Needed	Product Concept Developm ent Group of Product Managem ent Departme nt	105	US	√

Appendix E: Knowledge in the Collaborating Company

Enterprise Level

- Experiences of product managers and project managers
  - e.g. cost and budget estimation between generations
  - Similar product with a same development pattern
  - Similar management style and good for motivation
  - How can store their experiences in the KM Base
- Organisational principles and Rules (explicit)
  - 6 level product development principle
  - If several departments are cooperation, who is the leader and which one takes the main responsibility (cooperation matrix P29)
  - General principles and rules
- In-house product reviews and achievements, some specific knowledge about what this product bring and why
  - Previous product model, process, design model, framework
  - Previous product concept reports
- Supplier information and knowledge (explicit), what potential problems of each supplier, record of supplier, contractors, partners
  - Previous contract and contact with suppliers and contractors
  - Evaluation of each supplier and contractor includes characteristics, problems found, advantages and limits, and analysis of potential problems
  - Using this knowledge can identify which partner is most appropriate in a specified circumstance
- Board decisions of in-house product, what and why
  - High decision and solution previous product development, e.g. Candy cancel or change? (mini van vs. utility passenger vehicle)
  - Budget increase or performance decrease?
  - Produce in or purchase out?
- Enterprise level meeting minutes and reports, video and audio
  - Large and broad
  - Small but high level
- Lessons learned in enterprise level, e.g.
  - unsuccessful product development,
  - some product development shut down during the process and why
  - what risks and why
  - how much waste of the unsuccessful product development
  - How to manage the lessons learned. It could be small or large, can be

anywhere, description is enough? How to manage it?

- Benefit indication what potential benefit for a new product: technique and marketing
  - What facts can lead benefits, and how
  - e.g. entry of a undeveloped market, like pick-up truck in China
  - barrier product for competitor in existing market
  - enhance product in existing market
- Problems address and possible solutions
  - Predicted problems and possible solutions e.g. based on each development stage and operator what problems will be met. based on miles or years. e.g. component not fit may be from in-house or suppliers
  - Unpredicted problems and possible solutions e.g. some problems always happens, solutions are hard to find. Therefore, to find similar problems or same pattern or same circumstance is important to solve the problems
- Standardisations
  - Product concept development report template
  - Product development process pattern or template
  - Product line template and simulation
  - Customer assemble template with coach or bus
- Processes Modelling
  - Product concept development report template
  - Product development process pattern or template
  - Product line template and simulation
- Customer feedback
  - Feedback questionnaires using one product and pick up track or heavy track
  - Questionnaires for investigation of potential customers
- Investigation of market
  - Questionnaires for investigation of existing market
  - Questionnaires for investigation of potential market
- Product model
  - Design model such as car body framework, inner decoration, exhaust
- Policy Analysis and results
  - Pollution with engine
  - Size of engine < 1.6L has encouragement from government such as less road tax and tax free → results may lead change the size of engine of existing product.

Department Level (Strategic Development Department)

- Experiences of staff of this level, such as department leader, key staff of each

department function,
    - e.g. strategy analysis
- Experiences of expertises
- Lessons learned
- In-house product knowledge supports
- Communication records between departments
- Problems address and possible solutions
- Key department functions' explanations and key principle and rules
- Department level Meetings minutes and reports, video and audio
- Standardisations
- Relevant process and simulations
- Main function implementation and application

Department Level (Product Management Department)

- Experiences of staff of this level, such as department leader, key staff of each department function,
    - e.g. product concept process
- Experiences of expertises
    - e.g. previous planning of product development
- Lessons learned
- In-house product knowledge supports
- Communication records between departments
- Problems address and possible solutions
- Key department functions' explanations and key principle and rules
- Department level Meetings minutes and reports, video and audio
- Standardisations
- Relevant process and simulations
- Main function implementation and application

Department Level (Product Design Department)

- Experiences of staff of this level, such as department leader, key staff of each department function,
    - e.g. main designer of car body framework
- Experiences of expertises
    - Inner decorated component design
- Lessons learned
- In-house product knowledge supports
- Communication records between departments
- Problems address and possible solutions
- Key department functions' explanations and key principle and rules
- Department level Meetings minutes and reports, video and audio

- Standardisations
- Relevant process and simulations
- Main function implementation and application

Department Level (Product Manufacturing Department)

- Experiences of staff of this level, such as department leader, key staff of each department function,
    - e.g. Manufacturing process controlling
- Experiences of expertises
    - The sequence of manufacturing positions
- Lessons learned
- In-house product knowledge supports
- Communication records between departments
- Problems address and possible solutions
- Key department functions' explanations and key principle and rules
- Department level Meetings minutes and reports, video and audio
- Standardisations
- Relevant process and simulations
- Main function implementation and application

Department Level (Marketing and Sales Department)

- Experiences of staff of this level, such as department leader, key staff of each department function,
    - e.g. Product Positioning
- Experiences of expertises
    - e.g. Promotion
- Lessons learned
- In-house product knowledge supports
- Communication records between departments
- Problems address and possible solutions
- Key department functions' explanations and key principle and rules
- Department level Meetings minutes and reports, video and audio
- Standardisations
- Relevant process and simulations
- Main function implementation and application

Group Level – Product Concept Group

- Generated product concept report
- Relevant previous product concept report

- Key group principle and rules
- Relevant supporting process and models
- Experiences of staff of this level, such as group leader, key staff of each group
- Experiences of expertises
- Lessons learned
- In-house product knowledge supports
- Communication records between departments and groups vertically and horizontally
- Problems address and possible solutions
- Group level Meetings minutes and reports, video and audio
- Standardisations

Appendix F: Functional Requirements

The Wanted users' requirements are not converted to Functional Requirements

ID	Content	Enhanced based on Gaps	Category (NE/US/WT)	Functional Requirements
Enterprise Level				
UR 01	Product knowledge general management, such as store, search, invoke, integrate, share and etc	Specially focus on tacit knowledge: Lessons learned, personal experiences. Some Necessary Explicit knowledge: Standardisations, principles and rules and policies	NE	ID: FC_RE01, Definition: a function or meta- function to manage and control information /knowledge in the Knowledge Management System; Capability: to manage both explicit and tacit knowledge; Possible Functions required: Store, Search, Invoke, Integrate, Share.
UR 02	To manage the product Development process and procedure	the problems encountered during development process and the methods used to fix the problems	NE	ID: FC_RE02, Definition: a function or meta-function to model and manage processes; Capability: Modelling and Managing product development process, and possible to provide potential problems and their solutions when the process is modelled; Function required: process model, invoke, generation, store, sub-functions may be required
UR 03	To Model and manage the developing product lifecycle	Not Needed	NE	ID: FC_RE03, Definition: a function or meta-function to model and manage product lifecycle; Capability: Modelling and Managing product lifecycle; Function required: process model, invoke, generation, store, sub-functions may be

				required
UR 04	To ensure the potential benefit indication about the new product	New	NE	ID: FC_RE04, Definition: a function to find or search relevant knowledge in knowledge base in order to provide prediction of the potential benefits based on existing knowledge; Capability: Invoke and search knowledge in the knowledge base; Function required: knowledge search and invoke, store;
UR 05	To manage meeting Munities of each level	New	NE	ID: FC_RE05, Definition: a function to manage meeting munities; Capability: store, Invoke and search relevant knowledge in the knowledge base; Function required: knowledge search, store, invoke;
UR 06	To plan and manage human resource based on functions and positions in the process and production line	Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners	NE	ID: FC_RE06, Definition: a function or meta-function to plan and manage human resource in the production line; Capability: Modelling, Planning and Managing human resource in the production line, including knowledge about suppliers, contractors and partners; Function required: process model, invoke, generation, store, sub-functions may be required

UR 07	To estimate cost and Control budget of the product development project	Not Needed	NE	ID: FC_RE07, Definition: a function to find or search relevant knowledge in knowledge base in order to estimate cost and control budget based on existing knowledge; Capability: Invoke and search knowledge in the knowledge base; Function required: knowledge search and invoke, store;
UR 08	To manage the Board decisions which are based on the existing in-house product knowledge	New	NE	ID: FC_RE08, Definition: a function to manage board decision based on the existing in-house product knowledge; Capability: store, Invoke and search relevant knowledge in the knowledge base; Function required: knowledge search, store, invoke;
UR 09	To simulate the main functional implementation and application of each department	New	NE	ID: FC_RE09, Definition: To simulate the main functional implementation and application of each department
UR 10	To set up check points and milestones in the product development process	In-house product reviews and achievements, some specific knowledge about what this product can bring and why	NE	ID: FC_RE10, Definition: a function to set up check points, constraints and milestones based on the existing knowledge; Capability: store, Invoke and search relevant knowledge in the knowledge base; Function required: knowledge search, store, invoke;
UR 11	To classify functions of departments and allocate tasks of each department	Not Needed	NE	ID: FC_RE11, Definition: To simulate the main functional implementation and application of each department

	and group			
UR 12	To obtain the relevant product knowledge of benchmarking and previous projects	Not Needed	NE	ID: FC_RE12, Definition: a function to invoke benchmarking knowledge; Capability: store, Invoke and search relevant knowledge in the knowledge base; Function required: knowledge search, store, invoke;
UR 13	To cooperate and communicate with departments and the chair board vertically and horizontally	Not Needed	NE	ID: FU_RE13; Definition: Cooperate and communicate with departments and the chair board vertically and horizontally o Possible functions required: § The system contains several communications such as: blogs, emails, online messages, system phone calls and online chatting software. § All these communications should be stored as knowledge. o Capability for managing all resources and knowledge during the communications. System can provide an environment for communicating in the collaborating company
UR 14	To identify the authorisation of the product knowledge	Not Needed	NE	ID: FC_RE12, Definition: a function to set up constraints when authorised knowledge is invoked; Capability: store, Invoke and search relevant

				knowledge in the knowledge base; Function required: knowledge search, store, invoke;
Department Level				
UR 15	Require the concept development module in order to create a standard for the concept development	Not Needed	NE	ID: FC_RE15, Definition: a function or meta-function to create and manage concept development; Capability: a standard for creating and managing product concept development; Function required: concept creating module, model, invoke, generation, store, sub-functions may be required
UR 16	Modelling the product developing process	Not Needed	NE	ID: FC_RE16, Definition: a function or meta-function to model and manage processes; Capability: Modelling and Managing product development process; Function required: process model, invoke, generation, store, sub-functions may be required
UR 17	Simulating process or design in process required	Not Needed	NE	ID: FC_RE17, Definition: a function or meta-function to model and manage processes; Capability: Modelling and Managing product development process; Function required: process model, invoke, generation, store, sub-functions may be required

UR 18	to create standard and template for customer interview and feedback, and benchmarking of competitors and potential markets	Not Needed	NE	ID: FC_RE18, Definition: a function or meta-function to create and manage template and standard for customer feedbacks; Capability: a standard for creating and managing customer feedbacks; Function required: template creating module, model, invoke, generation, store, sub-functions may be required
UR 19	To simulate Manufacturing organisational structure and process	Not Needed	NE	ID: FC_RE19, Definition: a function or meta-function to model and manage processes; Capability: Modelling and Managing product development process; Function required: process model, invoke, generation, store, sub-functions may be required
UR 20	optimising product development process, the ideal way is automatic optimisation with system	Not Needed	NE	ID: FC_RE20, Definition: a function or meta-function to model and manage processes, specially for optimising product process; Capability: Modelling and optimising product development process; Function required: process model, invoke, generation, store, sub-functions may be required
UR 21	benchmarking – with the similar product in the knowledge base including both in-house product and competitor's	Not Needed	NE	ID: FC_RE21, Definition: a function for benchmarking based on the existing in-house product knowledge; Capability: store, Invoke and search relevant knowledge in the

	product			knowledge base; Function required: knowledge search, store, invoke;
UR 22	The department can search and invoke the necessary knowledge when the department needs to research the market, customers and competitors	Not Needed	NE	ID: FC_RE22, Definition: a function or meta- function to allow department to assess the relevant authorised knowledge; Capability: to assess both explicit and tacit knowledge; Possible Functions required: Store, Search, Invoke, Integrate, Share.
UR 23	To manage the researches of markets and customers	Not Needed	NE	ID: FC_RE23, Definition: a function or meta- function to manage information /knowledge from customers and markets in the Knowledge Management System; Capability: to manage both explicit and tacit knowledge; Possible Functions required: Store, Search, Invoke, Integrate, Share.
UR 24	The system can note each changes when the product is processing in the product line	Not Needed	NE	ID: FC_RE24, Definition: a function to notice all relevant departments or groups for any changes of the processes; Capability: notice individuals for changes; Function required: communication notices through email, telephone, message, and etc

UR 25	To manage and plan the raw material and resource included tracking logistics	Not Needed	NE	ID: FC_RE25, Definition: a function or meta-function to plan and manage raw material and resources; Capability: Modelling, Planning and Managing raw material and resources, including tracking logistics; Function required: process model, invoke, generation, store, sub-functions may be required
UR 26	digitalise graphs and figures – paperless work	Not Needed	US	ID: FC_RE26; Definition: Digitalise graphs and figures – paperless work; The function requires the system digitalise all models, processes and other diagrams. It also requires these diagrams can be printed out when it is required. o It is not an independent function of the system. It should be digitalised by system itself
UR 27	hierarchical management with functions in the department	Not Needed	US	ID: FU_RE27; Definition: manage department vertically and horizontally, this function actually can be implemented based on whole KM system design
UR 28	To gather, manage and integrate macro information and policy	Not Needed	US	ID: FC_RE28, Definition: a function to manage and integrate macro information, particularly for regulation and policies; Capability: to manage both explicit and tacit knowledge; Possible Functions required: Store, Search, Invoke, Integrate, Share.

UR 29	Diagram can be double checked with the relevant departments or staffs before allocating	Not Needed	US	ID: FC_RE29, Definition: a function to double check diagrams, models and processes with all relevant departments or groups; Capability: sign and release by relevant people; Function required: communication notices through email, telephone, message, and etc
UR 30	simultaneity design and development with other departments and groups	Not Needed	US	ID: FC_RE30, Definition: a platform for simultaneity design and communication between departments and groups; Capability: platform and modules; Function required: platform and modules
UR 31	To Manage industry research and analysis report	Not Needed	US	ID: FC_RE31, Definition: a function or meta- function to manage industrial research and analysis reports in Knowledge Management System; Capability: to manage both explicit and tacit knowledge; Possible Functions required: Store, Search, Invoke, Integrate, Share.
UR 32	large knowledge based to store the graphs and figure	Not Needed	US	ID: FC_RE32, Definition: a function or meta- function to manage meta knowledge or information about knowledge in Knowledge Management System; Capability: to manage both explicit and tacit knowledge; Possible Functions required: Store, Search, Invoke, Integrate, Share.

UR 33	multiple departments collaboration especially in product concept design	Not Needed	US	ID: FC_RE33, Definition: a platform for simultaneity design and communication between departments and groups; Capability: platform and modules; Function required: platform and modules
UR 34	visibility of the resource and controlling	Not Needed	WT	
UR 35	Needs more improvement in usability particularly in user interface and its menus, Unclear system structure, input inconvenient, lack communication, hard to search	Not Needed	WT	
UR 36	Planning simultaneity integration	Not Needed	WT	
UR 37	Diagram can be printed as the specific requirements	Not Needed	WT	
UR 38	Knowledge base is refreshed and shared rapidly	Not Needed	WT	
UR 39	OA should be integrated with KM System	Not Needed	WT	

UR 40	communication and logical link with different product developing periods	Not Needed	WT	
UR 41	Double checks with message and phone call	Not Needed	WT	
UR 42	Sockets are not match perfectly among sub-systems	Not Needed	WT	
UR 43	System integration – too many small systems in department, almost every departments has an individual system	Not Needed	WT	
Group Level				
UR 44	Develop a common framework for product concept development as a standard	Not Needed	NE	ID: FC_RE44, Definition: a function or meta-function to create and manage concept development; Capability: a standard for creating and managing product concept development; Function required: concept creating module, model, invoke, generation, store, sub-functions may be required

UR 45	Resource planning includes material planning, supplier, logistics	Not Needed	NE	ID: FC_RE45, Definition: a function or meta-function to plan and manage resource, material, suppliers and logistics; Capability: Modelling, Planning and Managing resource, material, suppliers and logistics in the production line, including knowledge about suppliers, contractors and partners; Function required: process model, invoke, generation, store, sub-functions may be required
UR 46	Benchmarking both in-house and outside product concept based on the system	Not Needed	US	ID: FC_RE46, Definition: a function to invoke benchmarking knowledge, particularly for product concept development; Capability: store, Invoke and search relevant knowledge in the knowledge base; Function required: knowledge search, store, invoke;
UR 47	Create a standard for product concept development process	Not Needed	US	ID: FC_RE47, Definition: a function or meta-function to create and manage concept development; Capability: a standard for creating and managing product concept development; Function required: concept creating module, model, invoke, generation, store, sub-functions may be required

UR 48	Require a common platform for better developing and communicating	Not Needed	US	ID: FU_RE48; Definition: Cooperate and communicate with departments and the chair board vertically and horizontally o Possible functions required: § The system contains several communications such as: blogs, emails, online messages, system phone calls and online chatting software. § All these communications should be stored as knowledge. o Capability for managing all resources and knowledge during the communications. System can provide an environment for communicating in the collaborating company

Appendix G: List of Functional Specifications after Benchmarking in the Product Management Department

Enterprise Level

- Knowledge Management system basic functional requirement (UR01)
    - Possible functions required: Store, Search, Invoke, Integrate, Share
    - Capability for manage tacit knowledge such as lessons learned, personal experiences, and etc.
    - Capability for necessary explicit knowledge: standardizations, principles, rules and policies
    - Category: necessary
- Modelling Process including each position and its resource of process and product line (UR02)
    - Possible functions required: Process Modelling, sub-functions might required
    - Capability for manage all of resources in the process including: supplier's records and resources of positions
    - Category: necessary
- To plan and manage human resource based on functions and positions in the process and production line (UR06)
    - ID: FC_RE06, Definition: a function or meta-function to plan and manage human resource in the production line;
    - Capability: Modelling, Planning and Managing human resource in the production line, including knowledge about suppliers, contractors and partners;
    - Function required: process model, invoke, generation, store, sub-functions may be required
- Cooperate and communicate with departments and the chair board vertically and horizontally(UR13)
    - Possible functions required:
        - The system contains several communications such as: blogs, emails, online messages, system phone calls and online chatting software.
        - All these communications should be stored as knowledge.

    - Capability for managing all resources and knowledge during the communications. System can provide an environment for communicating in the collaborating company
    - Category: necessary
- Manage meeting Munities of each level(UR05)
    - Possible functions required:
        - The main system functions are same as the UR01, i.e., it is knowledge management functions. However it is tacit knowledge
        - It requires that there is an interface which allows users to write meeting munities through the system.
        - The capability of this interface should satisfy most kinds of knowledge, e.g. models, process, text descriptions, audio and video.
    - Capability for recording every meeting contents in the collaborating company, especially in the Product Management Department.
    - Category: necessary
- Obtain the relevant product knowledge of benchmarking and previous projects (UR12)
    - Possible functions required:
        - The main system functions are based on the function UR01, i.e., it is knowledge management functions. The knowledge invoking and search functions are used to satisfy this function.
        - It requires that there is an individual interface which allows users to find the relevant knowledge, particularly for Planning and Product concept development.
        - The capability of this interface should satisfy most kinds of knowledge, e.g. models, process, text descriptions, audio and video.
    - Capability for seeking and invoking every relevant knowledge based on inputted conditions in the collaborating company, especially in the Product Management Department.
    - Category: necessary
- Identify the authorisation of the product knowledge (UR14)
    - Possible functions required:
        - The main system function is relevant with the UR01, i.e., it is knowledge management functions.
        - It requires to authorise knowledge with relevant users when the knowledge is searched and invoked.
    - Capability for authoring knowledge in each level of the collaborating company, especially in the Product Management Department.
    - Category: necessary

Department Level

- Digitalise graphs and figures – paperless work (UR26)
  - The function requires the system digitalise all models, processes and other diagrams. It also requires these diagrams can be printed out when it is required.
  - It is not an independent function of the system. It should be digitalised by system itself
  - Category: necessary
- Require the concept development module in order to create a standard for the concept development (UR16)
  - This function requires an independent interface for product concept development
  - Possible functions required:
    - A template for product concept development, such as a concept development report
  - Capability for the development of product concept in the Product Management Department through the systems.
  - Category: necessary
- Benchmarking – with the similar product in the knowledge base including both in-house product and competitor's product (UR22)
  - Possible functions required:
    - The main system functions are based on the function UR01, i.e., it is knowledge management functions. The knowledge invoking and search functions are used to satisfy this function.
    - It requires that there is an individual interface which allows users to find the relevant knowledge, particularly for Planning and Product concept development.
    - The capability of this interface should satisfy most kinds of knowledge, e.g. models, process, text descriptions, audio and video.
    - This function is similar to UR13, It might be developed as a part of UR13
  - Capability for seeking and invoking every relevant knowledge based on inputted conditions in the collaborating company, especially in the Product Management Department.
  - Category: necessary
- visibility of the resource and controlling (UR34)
  - It will not be contained in the system yet based on its low priority
  - Category: WT (low priority)

Group Level

- Develop a common framework for product concept development as a standard (UR44)
    - It is similar to UR16, it needs more details of product concept development framework.
    - It can be sub-functions of UR16
    - Category: necessary

- Resource planning includes material planning, supplier, logistics (UR45)
    - It can be a extensional function of the UR02
    - Possible functions required: Process Modelling, sub-functions might required
    - Capability for Planning and Managing all resources in the process including: supplier's records and resources of positions, and logistics
    - Category: necessary

- Benchmarking both in-house and outside product concept based on the system (UR46)
    - Possible functions required:
        - The main system functions are based on the function UR01, i.e., it is knowledge management functions. The knowledge invoking and search functions are used to satisfy this function.
        - It requires that there is an individual interface which allows users to find the relevant knowledge, particularly for Planning and Product concept development.
        - The capability of this interface should satisfy most kinds of knowledge, e.g. models, process, text descriptions, audio and video.
        - This function is similar to UR13, It might be developed as a part of UR13
    - Capability for seeking and invoking every relevant knowledge based on inputted conditions in the collaborating company, especially in the Product Management Department.
    - Category: Useful
- Create a standard for product concept development process (UR47)
    - It is a recruitment of UR02, but it is more specific for product concept development
    - Possible functions required: Process Modelling, sub-functions might required
    - Capability for manage all of resources in the process including: supplier's records and resources of positions
    - Category: Useful
- Require a common platform for better developing and communicating

  - Refer to UR03
  - Possible functions required:
    - The system contains several communications such as: blogs, emails, online messages, system phone calls and online chatting software.
    - All these communications should be stored as knowledge.
  - Capability for managing all resources and knowledge during the communications. System can provide an environment for communicating in the collaborating company
  - Category: Useful

Additional system specifications from benchmarking with other systems

- A modelling techniques tool set
  - Definition: Multi modelling techniques can be built in the system, such as IDEF, UML, ER-Relationship, PERT, Design Roadmap, and etc.
  - Possible related requirements: UR02, UR04, UR17
  - Category: Useful
- Product process is design and developed simultaneously
  - Possible functions: platform required
  - Related to UR02 "Simulate Process including each position and its resource of process and product line"
  - Category: Useful

Appendix H: Detailed System Specification System Specification

Introduction

Definition

The section introduces the system specification of knowledge management system in the collaborating company to its readers.

Objectives

- Provide a system overview of the knowledge management system in collaborating company including definition, goals, benefits and constraints
- To formally specify its associated:
  Functional requirements
  Suggested system specification

Intended Audiences

- The methodology designer
- System developers of the collaborating company
- Knowledge users from the collaborating company, including: product manager, department leaders, expert, employees.

Specification Overview

This specification is a sample for the knowledge management system in the collaborating company. One department is implemented using the developed methodology.

System overview

Definitions

The knowledge management system will provide a comprehensive environment to manage enterprise knowledge about product development in the collaborating company

Business goal

The knowledge management system will manage the knowledge of product development systematically. It improves the current situation of messed knowledge management in the collaborating company.

Business benefits

- Manage the knowledge in product development systematically, in order to avoid extra cost of unmanaged knowledge
- Improve the efficiency of daily work in the Product Development Department

Functional Requirements

The Product Management Department

Definition

The department mainly charges the planning stage in the business process

Responsibilities

- Mainly charges planning for product development including resource planning, product line planning and process planning
- Developing the product concept

List of Requirements

Enterprise Level

- Knowledge Management system basic functional requirement (UR01)
  - Possible functions required: Store, Search, Invoke, Integrate, Share
  - Capability for manage tacit knowledge such as lessons learned, personal experiences, and etc.
  - Capability for necessary explicit knowledge: standardizations, principles, rules and policies
  - Category: necessary
- Modelling Process including each position and its resource of process and product line (UR02)
  - Possible functions required: Process Modelling, sub-functions might required
  - Capability for manage all of resources in the process including: supplier's records and resources of positions
  - Category: necessary
- To plan and manage human resource based on functions and positions in the process and production line (UR06)
  - ID: FC_RE06, Definition: a function or meta-function to plan and manage human resource in the production line;
  - Capability: Modelling, Planning and Managing human resource in the production line, including knowledge about suppliers, contractors and partners;
  - Function required: process model, invoke, generation, store, sub-functions may be required
- Cooperate and communicate with departments and the chair board vertically and horizontally(UR13)
  - Possible functions required:
    - The system contains several communications such as: blogs, emails, online messages, system phone calls and online chatting software.
    - All these communications should be stored as knowledge.

    - Capability for managing all resources and knowledge during the communications. System can provide an environment for communicating in the collaborating company
    - Category: necessary
- Manage meeting Munities of each level(UR05)
    - Possible functions required:
        - The main system functions are same as the UR01, i.e., it is knowledge management functions. However it is tacit knowledge
        - It requires that there is an interface which allows users to write meeting munities through the system.
        - The capability of this interface should satisfy most kinds of knowledge, e.g. models, process, text descriptions, audio and video.
    - Capability for recording every meeting contents in the collaborating company, especially in the Product Management Department.
    - Category: necessary
- Obtain the relevant product knowledge of benchmarking and previous projects (UR12)
    - Possible functions required:
        - The main system functions are based on the function UR01, i.e., it is knowledge management functions. The knowledge invoking and search functions are used to satisfy this function.
        - It requires that there is an individual interface which allows users to find the relevant knowledge, particularly for Planning and Product concept development.
        - The capability of this interface should satisfy most kinds of knowledge, e.g. models, process, text descriptions, audio and video.
    - Capability for seeking and invoking every relevant knowledge based on inputted conditions in the collaborating company, especially in the Product Management Department.
    - Category: necessary
- Identify the authorisation of the product knowledge (UR14)
    - Possible functions required:
        - The main system function is relevant with the UR01, i.e., it is knowledge management functions.
        - It requires to authorise knowledge with relevant users when the knowledge is searched and invoked.
    - Capability for authoring knowledge in each level of the collaborating company, especially in the Product Management Department.
    - Category: necessary

Department Level

- Digitalise graphs and figures – paperless work (UR26)
    - The function requires the system digitalise all models, processes and other diagrams. It also requires these diagrams can be printed out when it is required.
    - It is not an independent function of the system. It should be digitalised by system itself
    - Category: necessary
- Require the concept development module in order to create a standard for the concept development (UR16)
    - This function requires an independent interface for product concept development
    - Possible functions required:
        - A template for product concept development, such as a concept development report
    - Capability for the development of product concept in the Product Management Department through the systems.
    - Category: necessary
- Benchmarking – with the similar product in the knowledge base including both in-house product and competitor's product (UR22)
    - Possible functions required:
        - The main system functions are based on the function UR01, i.e., it is knowledge management functions. The knowledge invoking and search functions are used to satisfy this function.
        - It requires that there is an individual interface which allows users to find the relevant knowledge, particularly for Planning and Product concept development.
        - The capability of this interface should satisfy most kinds of knowledge, e.g. models, process, text descriptions, audio and video.
        - This function is similar to UR13, It might be developed as a part of UR13
    - Capability for seeking and invoking every relevant knowledge based on inputted conditions in the collaborating company, especially in the Product Management Department.
    - Category: necessary
- visibility of the resource and controlling (UR34)
    - It will not be contained in the system yet based on its low priority
    - Category: WT (low priority)

Group Level

- Develop a common framework for product concept development as a standard (UR44)
    - It is similar to UR16, it needs more details of product concept development framework.
    - It can be sub-functions of UR16
    - Category: necessary

- Resource planning includes material planning, supplier, logistics (UR45)
    - It can be a extensional function of the UR02
    - Possible functions required: Process Modelling, sub-functions might required
    - Capability for Planning and Managing all resources in the process including: supplier's records and resources of positions, and logistics
    - Category: necessary

- Benchmarking both in-house and outside product concept based on the system (UR46)
    - Possible functions required:
        - The main system functions are based on the function UR01, i.e., it is knowledge management functions. The knowledge invoking and search functions are used to satisfy this function.
        - It requires that there is an individual interface which allows users to find the relevant knowledge, particularly for Planning and Product concept development.
        - The capability of this interface should satisfy most kinds of knowledge, e.g. models, process, text descriptions, audio and video.
        - This function is similar to UR13, It might be developed as a part of UR13
    - Capability for seeking and invoking every relevant knowledge based on inputted conditions in the collaborating company, especially in the Product Management Department.
    - Category: Useful
- Create a standard for product concept development process (UR47)
    - It is a recruitment of UR02, but it is more specific for product concept development
    - Possible functions required: Process Modelling, sub-functions might required
    - Capability for manage all of resources in the process including: supplier's records and resources of positions
    - Category: Useful
- Require a common platform for better developing and communicating

- Refer to UR03
- Possible functions required:
    - The system contains several communications such as: blogs, emails, online messages, system phone calls and online chatting software.
    - All these communications should be stored as knowledge.
- Capability for managing all resources and knowledge during the communications. System can provide an environment for communicating in the collaborating company
- Category: Useful

Additional system specifications from benchmarking with other systems

- A modelling techniques tool set
    - Definition: Multi modelling techniques can be built in the system, such as IDEF, UML, ER-Relationship, PERT, Design Roadmap, and etc.
    - Possible related requirements: UR02, UR04, UR17
    - Category: Useful
- Product process is design and developed simultaneously
    - Possible functions: platform required
    - Related to UR02 "Simulate Process including each position and its resource of process and product line"
    - Category: Useful

Constraints

System Constraints

N/a

Business Rules

- Business objectives
- Budget
- Labours
- Size of the company

Data and Content Constraints

- Explicit knowledge
- Tacit knowledge
- Current existing knowledge
- Current information and data

- Digitalised models, processes and other diagrams

Databases

None

Hardware Constraints

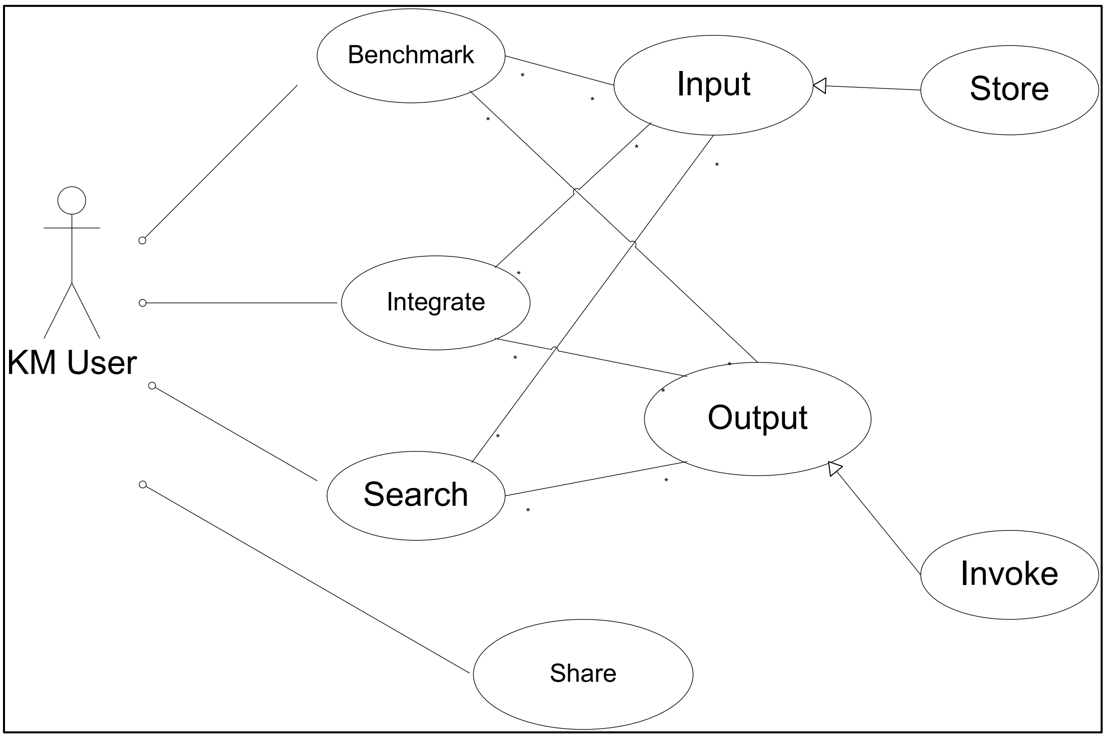
None

Software Constraints

None

System Design Specification

Knowledge Management System Basics Functions:



Final User’s Requirement ID.
UR01

Functional Requirement ID
FC_RE01

Descriptions of the Initial User’s Requirement

1. A set of functions for basic knowledge management system
2. Integrated all user's requirements about knowledge basic

Overview

According the user's requirements, there are 5 main users' requirements and 1 suggested function need to be applied by the basic knowledge functions

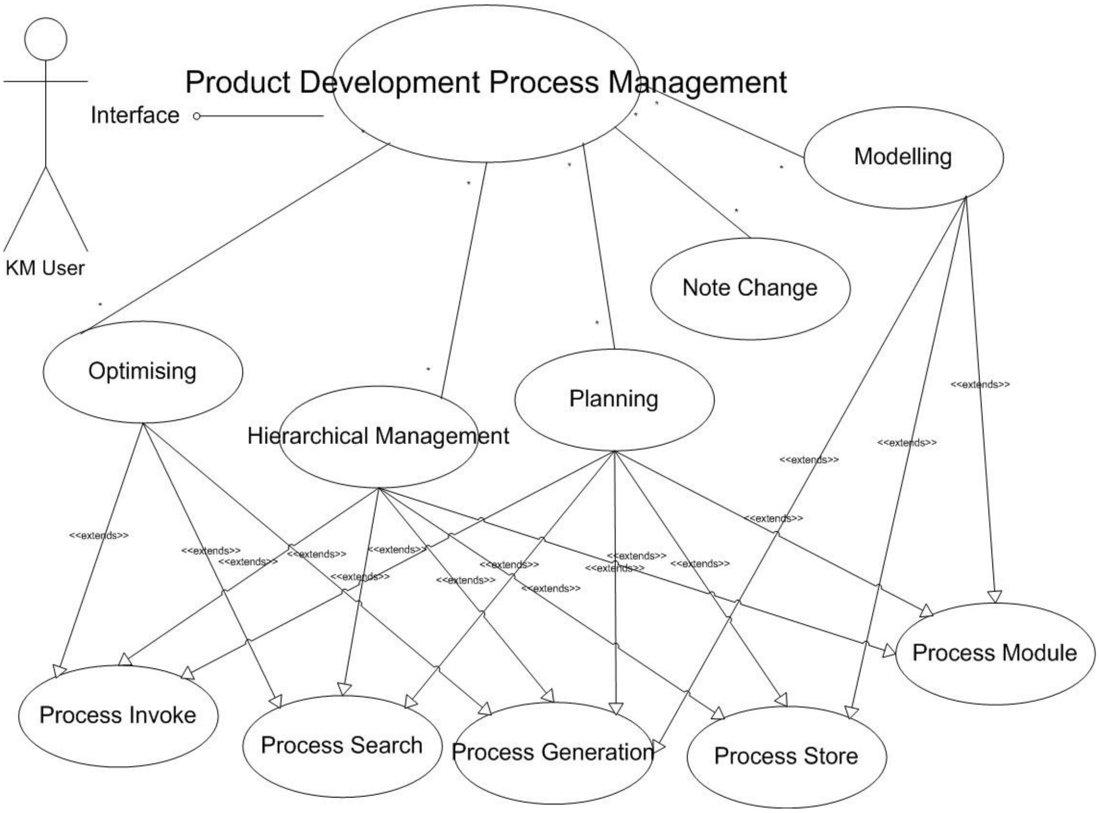
Functions (with Reference NO):

Knowledge store (F_URKB_1)
Knowledge search(F_URKB_2)
Knowledge invoke(F_URKB_3)
Knowledge Share (F_URKB_4)
Knowledge Integrate (can be implemented through others)
Knowledge Benchmark (can be implemented through others)

Stakeholder Needs List

All of stakeholders

Product Development Process (Product_Development_Process)



Initial User's Requirement NO.

UR02,UR04

Descriptions of the Initial User's Requirement

-Integration of all requirements of the product process management

- the problems encountered during development process and the methods used to fix the problems

-Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners

Overview

According the list of user's requirements, there are several user's requirements about the product development process. The integrated user's requirements includes the functions which shown as the left figure

Functions (with Reference NO):

Process Modelling (F_UR_PROCESS1)

Low-level sub-functions

- Process Module
- Process Store
- Process Generation

Process Optimising(F_UR_PROCESS2)

Low-level sub-functions

- Process Search
- Process Store
- Process Invoke

Process Planning (F_UR_PROCESS3)

Low-level sub-functions

- Process Module
- Process Store
- Process Generation
- Process Search
- Process Invoke

Hierarchical Management (F_UR_PROCESS4)

Low-level sub-functions

- Process Module
- Process Store
- Process Generation
- Process Search
- Process Invoke

Note Change (F_UR_PROCESS5)

Stakeholder Needs List

Product manager, Product Management Department, Design Department, Manufacturing Department

Detailed Functions Development

Owners and List of Contacts

Name	Email	Phone	Role
Mr Xing	x.xing@xxxx.com	0086-10-887456xx	Head of Product

			Management Department
Mr Shao			Associate of Product Management Department
Mr Ye			Head of IT Department
Mr Zhang			Product Manager
Mr Chen			Group Leader in Product Management Department
Mr Li			Group Leader in Product Management Department

Signoffs

Phase	Name	Date	Signature
Functional Specifications	Pengcheng Zhang	xx/xx/xx	

Revision History

Date	Reason for change(s)	Author(s)
04/12/2009	*First Draft*	*P. Zhang*
06/06/2010	Version after Evaluation	p. Zhang

Summary

This document describes the features and timeframe desired for IS/KM system for the Product Management Department in the Collaborating Company

Project Goals, Justification, and Success Criteria

Project Goals

The aim of this project is to verify the developed methodology for knowledge system design to support new product development in the Product Management Department in the Collaborating Company

Justification

This project is used to verify and evaluate the developed methodology is suitable in a real industrial environment.

Success Criteria

The success criteria is to develop the functional specification and evaluate the functional specification with the Product Management Department and other relevant people

Functions Design about the Functional Specification UR01

Definition

UR01 is basic system function for Knowledge Management

Final User's Requirement ID.

UR01, UR08, UR13, UR14, UR22, UR46

Descriptions of the Initial User's Requirement

1. A set of functions for basic knowledge management system
2. Integrated all user's requirements about knowledge basic

Overview

According the user's requirements, there are 5 main users' requirements and 1 suggested function need to be applied by the basic knowledge functions

Functions (with Reference NO):

Knowledge store (F_URKB_1)

Knowledge search (F_URKB_2)

Knowledge invoke (F_URKB_3)

Knowledge Share (F_URKB_4)

Knowledge Integrate (can be implemented through others)

Knowledge Benchmark (can be implemented through others)

Knowledge Authorisation

Stakeholder Needs List

All of stakeholders

All Possible Functions

Function ID:

F_URKB_1

Function Name:

Knowledge Store

Description:

This is a function which allow the users can input knowledge through the interface

Component Functions :

N/a

Stakeholder Needs List:

All of stakeholders

Function ID:

F_URKB_2

Function Name:

Knowledge Search

Description:

This is a function which allow the users can search knowledge in knowledge base through the interface

Component Functions:

N/a

Stakeholder Needs List:

All of stakeholders

Function ID:

F_URKB_3

Function Name:

Knowledge Invoke

Description:

This is a function which allow the users can invoke the correct knowledge when it is required

Component Functions:

N/a

Stakeholder Needs List:

All of stakeholders

Function ID:

F_URKB_4

Function Name:

Knowledge Share

Description:

This is a function which allow knowledge to be share in the system

Component Functions:

N/a

Stakeholder Needs List:

All of stakeholders

Function ID:

F_URKB_5

Function Name:

Knowledge Authorisation

Description:

This is a function which allow knowledge can be accessed by authorised users

Component Functions:

N/a

Stakeholder Needs List:

All of stakeholders

Design Functions for Modelling Process

Initial User's Requirement NO:

UR02, UR04, UR45, UR47

Descriptions of the Initial User's Requirement:

-Integration of all requirements of the product process management

- the problems encountered during development process and the methods used to fix the problems

-Supplier information and knowledge, what potential problems of each supplier, record of supplier, contractors, partners

Overview

According the list of user's requirements, there are several user's requirements about the product development process. The the integrated user's requirements includes the functions which shown as the left figure

Functions (with Reference NO):

Process Modelling (F_UR_PROCESS1)

Low-level sub-functions

- Process Module
- Process Store
- Process Generation

Process Optimising(F_UR_PROCESS2)

Low-level sub-functions

- Process Search
- Process Store
- Process Invoke

Process Planning (F_UR_PROCESS3)

Low-level sub-functions

- Process Module

- Process Store
- Process Generation
- Process Search
- Process Invoke

Hierarchical Management (F_UR_PROCESS4)

Low-level sub-functions

- Process Module
- Process Store
- Process Generation
- Process Search
- Process Invoke

Note Change (F_UR_PROCESS5)

Stakeholder Needs List

Product manager, Product Management Department, Design Department, Manufacturing Department

All Possible Functions and Sub-functions

Function ID:

F_UR_PROCESS1

Function Name:

Process Modelling

Description:

This is a high level function which allow the users can model the process through the KM system

Component Functions:

Process Module

Process Producing

Process Store

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Function ID:

F_UR_PROCESS2

Function Name:

Process Optimising

Description:

This is a high level function which allow the users can Optimising the process through the KM system

Component Functions:

Process Search

Process Invoke

Process Producing

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Function ID:
F_UR_PROCESS3
Function Name:
Process Planning
Description:
This is a high function which allow the users can plan the process through the KM system
Component Functions:
Process Module
Process Producing
Process Store
Process Search
Process Invoke
Stakeholder Needs List:
Product manager, Product Management Department, Design Department, Manufacturing Department

Function ID:
F_UR_PROCESS4
Function Name:
Process Planning
Description:
This is a high function which allow the users can manage the process through the KM system based on different departments and groups in different levels
Component Functions:
Process Module
Process Producing
Process Store
Process Search
Process Invoke
Stakeholder Needs List:
Product manager, Product Management Department, Design Department, Manufacturing Department

Function ID:
F_UR_PROCESS5
Function Name:
Note Change
Description:
This is a function which allow the users can note the process has been already changed through the KM system to other users
Component Functions:

n/a

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Sub-Functions

Function ID:

F_UR_PROCESS1_1

Function Name:

Process Module

Description:

This is a sub function which provides a module to users and allow users can producing a process

Component Functions:

n/a

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Related Functions:

F_UR_PROCESS1,F_UR_PROCESS3,F_UR_PROCESS4

Function ID:

F_UR_PROCESS1_2

Function Name:

Process Producing

Description:

This is a sub function which allow users can producing a process which followed the process module

Component Functions:

n/a

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Related Functions:

F_UR_PROCESS1,F_UR_PROCESS2,F_UR_PROCESS3,F_UR_PROCESS4

Function ID:

F_UR_PROCESS1_3

Function Name:

Process Store

Description:

This is a sub function which allows users can store the produced process

Component Functions:

n/a

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Related Functions:

F_UR_PROCESS1,F_UR_PROCESS3,F_UR_PROCESS4

Function ID:

F_UR_PROCESS2_1

Function Name:

ProcessSearchF_UR_PROCESS3,F_UR_PROCESS4

Description:

This is a sub function which allow users can search a process

Component Functions:

n/a

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Related Functions:

F_UR_PROCESS2,F_UR_PROCESS3,F_UR_PROCESS4

Function ID:

F_UR_PROCESS2_2

Function Name:

Process Invoke

Description:

This is a sub function allow users can invoke a process

Component Functions:

n/a

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Related Functions:

F_UR_PROCESS2,F_UR_PROCESS3,F_UR_PROCESS4

Design Function for Communication

Initial User's Requirement NO.

UR03, UR48

Descriptions of the Initial User's Requirement

Cooperate and communicate with departments and the chair board vertically and horizontally.

Overview

Its capability manages all resources and knowledge during the communications. System can provide an environment for communicating in the collaborating company

Functions (with Reference NO):

Communication (F_UR__COM01)

blogs,
emails,
online messages,
system phone calls
online chatting software.

All these communications should be stored as knowledge.

Refer to KM_Basics

Stakeholder Needs List

Product manager, Product Management Department, Design Department, Manufacturing Department

Functions

Function ID:

F_UR__COM01

Function Name:

Communication

Description:

This is a function allow users to communicate each other through the system

Component Functions:

n/a

Stakeholder Needs List:

Product manager, Product Management Department, Design Department, Manufacturing Department

Related Functions:

blogs,
emails,
online messages,
system phone calls
online chatting software.

Function Design for Product Concept Development Module

Initial User's Requirement NO.

UR16, UR 44

Descriptions of the Initial User's Requirement

It is required to contain a concept development module to create a standard for the concept development

Overview

This function requires an independent interface for product concept development. A template is required for product concept development, such as a concept development report. Its capability is for the development of product concept in the Product Management Department through the systems.

Functions (with Reference NO):

Product Concept Template (F_UR_PCD01)

- A template for product concept development, such as a concept development report
  - Create
  - Insert
  - Knowledge Management Refer to KM_Basics, such as store, search, invoke

Stakeholder Needs List

Product Management Department

Functions

Function ID:

F_UR_PCD01

Function Name:

Template for the Development of Product Concept

Description:

This is a function allow users to create product concept based on a template as a standard through the system

Component Functions:

n/a

Stakeholder Needs List:

Product Management Department

Related Functions:

Create product concept

Security Requirements

It is confidential project in the collaborating company. The relevant knowledge and information only can be accessed by the Product Management Department and some other staff with authorisation.

Data Conversion Requirements

Explain any requirements in which we must convert data to import into the system. Detail the mapping needed from their system to the new system.

Performance and Response Time Requirements

15 concurrent users will be using the system in the Department. The requirement of response time is not specified.

Platform Dependent and Installation Requirements

None

Localisation Requirements

In the Collaborating Company in Beijing, China

Parallel Testing Requirements

Evaluating the functional specification with the staff of the Product Management Department and other relevant staff in the collaborating company such as staff from IT department

Cross System Interface Requirements

None

Data Archival, Backup, Recovery and Reporting Requirements

None