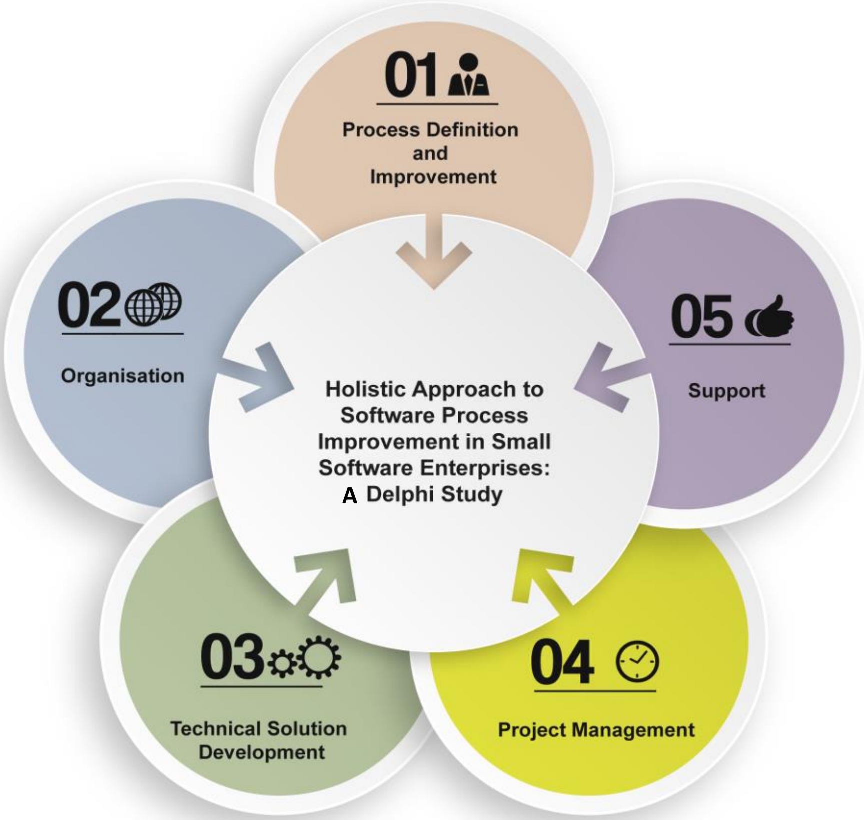
Holistic Approach to Software Process Improvement in Small Software Enterprises: A Delphi Study

Submitted by **Fatina H. Aweidah**



A thesis submitted to Auckland University of Technology in partial fulfilment of the requirements for the degree of

Master of Philosophy (MPhil)

August 2015

Department of Business Information Systems
Faculty of Business and Law

Supervisor: **Professor Felix B Tan**

ABSTRACT

For small software enterprises with 50 or fewer employees, using international standards to manage their software development projects has been a major challenge as these standards are more suited to large organisations. This has presented a challenge to small software enterprises (SSEs) since they operate in highly challenging and constantly changing environments and tend to perform their productive, administrative and management processes in a chaotic manner. There is agreement in the literature that SSEs need superior internal systems and structures to help them achieve their project objectives. Despite this, there is limited research into developing a process model that incorporates best practice and is suitable for SSEs.

The purpose of this research study is to develop a capability model with a set of essential practices for SSEs. The analysis of the literature of Software Process Improvement (SPI) and Dynamic Capabilities (DC) suggested the need for a wide array of capabilities that account for different aspects of the organisation and the external environment.

This research, therefore, is based on the premise that organisations can take a holistic multifaceted approach to SPI programmes. Sound organisational processes based on best practice are needed to improve their internal structures. Embedding dynamic capabilities within these processes will additionally enable the enterprise to address the business challenges in competitive markets.

An initial version of the proposed model was developed. This provided the necessary practices to help SSEs meet their perceived project outcomes. The model builds upon multiple software engineering standards, informed by the Dynamic Capability approach, and designed to be feasible for application in SSEs.

Using the Delphi technique as the methodology, this study sought the help of a group of experts to assess and enhance the proposed model by removing or adding practices to make it feasible for SSEs. A panel of nine members experienced in software process improvement and software engineering from different geographic regions (South Asia, North America and Australasia) responded to two Delphi rounds; their views about the

model were probed through qualitative analysis of members' feedback on the practices, supported by quantitative analysis using the interquartile range (IQR) and median measures.

The outcome of this research is a capability model that includes a set of basic and advanced practices, which represents the model that the participants have collectively approved. This model hopes to be an effective software development framework for SSEs. Further work is still required to verify the model using a case study research approach, in order to compare the Delphi findings against observed data.

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Fatina

Fatina H. Aweidah

25 August 2015

ACKNOWLEDGEMENT

This thesis is the result of almost four years of part-time study. I am very delighted that it has finally ended. During this period, a number of critical events took place in my life that delayed earlier completion of this work. During this time, I was surrounded with many supportive people; without them it may not have been possible to finish this work.

I would like to express my deepest gratitude to my supervisor, Professor Felix B Tan, for his guidance and patience, and for providing me with constructive comments and valuable insights to improve this work. His research expertise and optimism helped me overcome several obstacles and changed the direction of my research slightly.

I would also like to express my gratitude to Professor Stephen MacDonnell who guided and supervised my work in the early stages of this study. His enthusiasm, knowledge of software process improvement and experience in research helped me set the foundation for this study.

My thanks also go to all the special people in my life, especially my family to whom I am indebted for their willingness to accommodate to me and to help me achieve my goals. To my brothers and their families thank you all for your support and encouragement. To my wonderful mother, for being the source of inspiration - without you this would have been harder. Finally, I dedicate this work to my late father, who missed the opportunity to celebrate the completion of this milestone. Thank you for teaching me that the key to success is perseverance.

This research is approved by the Auckland University of Technology Ethics Committee on 5 September 2013 AUTEC Reference number 13/221.

ABBREVIATIONS

SPI	Software Process Improvement
SSE	Small Software Enterprises
SE	Small Enterprises
DC	Dynamic Capabilities
PD&I	Process Definition and Improvement Capability
ORG	Organisational Capability
PM	Project Management
TSD	Technical Solution Development
SUP	Support

CHAPTER 1 - INTRODUCTION

1.1 Research Topic

This thesis proposes a reference model for small software enterprises (SSEs) that is composed of a set of processes for five identified capabilities intended to guide SSEs to improve their processes to help them in attaining better project outcomes as well as improving the overall firm effectiveness and capabilities. The model is developed by integrating and harmonizing at high-level processes from number of Software Process Improvement (SPI) standards.

The acceptance of the model by nine software development consultants and practitioners is empirically assessed using a Delphi study. In addition to meeting the objectives of the research, the findings provided a number of insights on factors influencing process improvement initiatives in SSEs.

This chapter sets the foundation for this research project. Firstly, the background information about the disciplinary areas involved in this research is provided, and the significance and distinctiveness are established. Next, the research scope and aim are stated. Finally, the research questions and objectives are defined.

1.2 Introduction

There are several global trends having an impact on the modern business enterprise and the most critical among them is the advancement in information communication technologies which continues to transform the world into a single global village with a global economy Pleatsikas and Teece (2001)

Because of this integrated global market and the rapid changes in technology, the modern business enterprise is operating in an environment of discontinuous change (Haeckel, 2000; Pleatsikas & Teece, 2001). The online business dictionary defines this term as sudden change that threatens the traditional power structure because it drastically alters the way things are done. Research referring to this phenomenon describes it as hypercompetitive or high velocity environments experiencing major drastic and unforeseen changes in technological, competitive and regulatory domains (Barreto, 2010). This environment is not only threatening the firm's position in industry but also its survival. This global environment requires a different business model to deal with the associated uncertainty, instability and continuous market changes. It calls attention to the need for a strategic management approach that helps managers to understand both the internal and external environment in order to identify potential opportunities and threats and to ensure strategic competitiveness and good financial returns (Pleatsikas & Teece, 2001).

The dynamic capability literature suggests that in such hypercompetitive markets entrepreneurial capabilities may be as or more important than technical capabilities (Pleatsikas & Teece, 2001). Firms are in need of effective capabilities that will enable them to identify new business opportunities and changing customer needs as they happen, then react to these challenges fast and appropriately before the strategic significance is lost (Eisenhardt, 2000; Kivelä, 2007; Zollo & Winter, 1999). Therefore, effective strategic management requires continuous strategic planning and constant strategy reviews as opposed to an annual strategic review process. This strategic approach requires supporting processes and competencies to coordinate and manage the firm's internal resources in a particular way to adopt to changing customer and technological needs (Teece, 2007).

This trend presents particular challenges to small software enterprises (SSEs) which not only operate in high velocity environments, but also have additional challenges inherent in their nature (Eisenhardt, 2000). Evidently, there are other common challenges shared among all small enterprises (SEs) in general, which are believed to affect their performance. The participation and involvement of an owner-manager in all aspects of the business were highlighted in this regard (A Cater-Steel, 2004a; Mazzarol, Reboud, & Soutar, 2009). Moreover, SEs operate with comparatively little capital, have access to fewer resources, and operate in a rapidly changing environment (Kivelä, 2007) ; all these factors make them very vulnerable to changes in the external environment.

Practitioners and academics acknowledge that SSEs involved in the development of software experience the same challenges as small enterprises in other sectors, but additionally have challenges of their own. Researchers argue that SSEs perform their productive, administrative and management processes in a chaotic manner and deliver their software products in a culture of urgency (Zavala-Ruiz, 2008). Researchers identified issues for SSEs, among them the lack of an effective and structured approach to handling technical and management issues, the absence of formal and institutionalised practices, and inconsistency in the implementation of processes across the organisation (A Cater-Steel, 2001; C Laporte et al., 2005). Hence, SSEs are in need of process improvement programmes to guide them and improve their internal structures and strategic practices.

This research therefore argues that sound organisational processes based on best practice have positive impacts on the organisation's performance (Yu-Chih Liu, Chen, Chan, & Lie, 2008). Embedding dynamic capabilities within these processes will additionally enable the enterprise to address the business challenges within dynamic and hypercompetitive markets. With dynamic capabilities, the firm can deliver value to customers by developing a combination of organisation processes, human skills and technological innovations; these result in competencies that are difficult for competitors to imitate. Firms developing capabilities that competitors find difficult to imitate are likely to outperform competitors lacking those capabilities.

The main objective of this research is to develop a capability model for SSEs. The model draws on the SPI literature to define the required operational processes, the Dynamic Capability literature to enable the enterprise to survive in dynamic and hypercompetitive markets, and the literature of Management and Small Organisations, which is discussed in the next chapter, to gain insights into the challenges facing SEs in general.

1.3 Dynamic Capability Approach

The framework of Dynamic Capabilities (DC) as a source of competitive advantage is at the forefront of strategy research. Teece and Pisano (1994) promoted a strategic approach grounded in the resource-based view (RBV) that tends to direct managers to focus on creating distinctive and difficult-to-imitate strategic advantages and avoid the traditional strategy approaches that rely on rivalry games with competition and customers. The corner stone of DC is managerial and organisational processes that help the firm to effectively coordinate and deploy internal and external competencies to achieve congruence with the changing business environment (Teece, Pisano, & Shuen, 1997). Hence, some organisations are profitable not because of the strategic moves and counter moves they have used to influence competitors' positions in industry, but because they have developed efficient processes that help them identify opportunities for change, formulate a response and implement the change. The benefits that enterprises can obtain from dynamic capability depend on how well those DCs fit with the firm's internal and external environment. Consequently, the firm can increase productivity, lower cost or offer products of superior quality and is able to bring value to customers.

Teece et al. (1997) used the term 'Dynamic Capabilities' to stress the ability of the firm to utilise internal and external firm-specific competencies to address the changing business environment.

This study uses the definition proposed by Barreto (2010) and defines Dynamic Capabilities as a set of organisational capacities embedded in the organisation's formal and informal practices (processes and culture) to sense opportunities and threats, to make timely and market-oriented decisions, and to change its resource base.

1.4 Software Process Improvement Approach

Software Process Improvement is rooted in the software engineering and information systems disciplines. Research in this area is mainly concerned with professional practices (i.e. processes) related to the management and development of software products; there is an extensive body of knowledge and international standards promoting best practice. A number of international standards have influenced both the literature and practice.

Since the 1990s the software engineering community has developed a growing interest in SPI with the motivation to improve the quality of the software product by focusing on improving organisational processes to match best practice recommended by international standards. This set of international standards has evolved to provide practices accepted by industry and academics. These practices are informed by empirical research and industry feedback and can be utilised within an organisation or project. Examples of well-accepted and practised standards among the software development community are SPI and Capability Determination (ISO/IEC 15504), Capability Maturity Model Integration (CMMI), Project Management Body of Knowledge (PMBOK), Information Technology Infrastructure Library (ITIL), and others. The IT community refers to international standards developed for this purpose as SPI Standards.

1.5 Problem Definition

The introduction section highlighted that SSE need to pay attention to their managerial and operational processes and entrepreneurial practices. Furthermore, SSEs need to adopt dynamic capabilities to guide their operations in dynamic markets. Dynamic markets require new ways of thinking about strategies and capabilities that would enable SSEs to sense and exploit opportunities in addition to sensing threats and minimising their impact. To gain competitive advantage there is a need to understand customers and markets at project and firm level (Mathiassen & Vainio, 2007). To achieve this objective, the dynamic capabilities developed must therefore address the firm's internal and external environments. However, Mathiassen and Vainio (2007) reported that existing dynamic capability approaches applied by small software firms focus on particular customer projects, adopt a technological perspective and ignore interactions with other business dimensions and the external environment.

SPI literature also emphasises that a software enterprise requires the development of a wide array of capabilities that account for internal aspects of the organisation and the external environment, in addition to the technical engineering (Barafort, Di Renzo, & Merlan, 2002). Moreover, literature highlights that SPI standards have been criticised for not taking a holistic approach towards the organisation as an integrated social and business entity. For example, this was pointed out by Allison and Merali (2007) who argued that the majority of SPI standards focus on the technical and engineering processes and overlook the relationship between these and other external environmental and organisational factors. However, a number of other research studies emphasised the fact that engineering processes function best when matched with other internal and external aspects of the organisation such as human related factors, resources, culture, strategy and the market in which the organisation operates (Allison & Merali, 2007; Cattaneo, Fuggetta, & Sciuto, 2001; Kirk & MacDonell, 2009).

It is clear from the above discussion that there is an implicit agreement among SPI researchers that SPI standards are failing to address both internal and external aspects of the organisation. Hence, they are not guiding the organisation to take the right strategic direction to survive or grow in highly dynamic markets. Thus, organisations are in need of dynamic capabilities to fill this gap. As will be seen in Chapter 2, the literature review did not reveal any existing SPI research that draws on the theory of dynamic capabilities. Moreover, while the literature provided a number of frameworks to guide enterprises to define dynamic capabilities and to help them adapt to the changing market conditions (Eisenhardt, 2000 ; Kivelä, 2007), there are no reported framework that follow a comprehensive approach available yet for small software enterprises (Kivelä, 2007). Most importantly, there are no reported frameworks for SSEs that can help the firms to effectively respond to changes in customer demands, market opportunities, and technology options (Mathiassen & Vainio, 2007).

In an effort to take a more holistic approach, many organisations are implementing multiple frameworks to accommodate areas not addressed by the adopted international standard. The different SPI standards focus on a specific part of the business. For example, PMBOK focuses on project management practices, ITIL focuses on service management practices (Hochstein, Zarnekow, & Brenner, 2005), and ISO/IEC 12207's focus is technical engineering. However, by focusing on and improving one area of a business these models are `not helping organisations to achieve their perceived business objectives overall. Nevertheless, evidence indicates an increasing adoption of multiple frameworks simultaneously by organisations (A Cater-Steel, Tan, & Toleman, 2006). Popular combinations of frameworks have been reported such as CMMI-DEV and ISO/IEC 15504 or COBIT and CMMI (A Cater-Steel et al., 2006). A number of studies attempted to map and harmonize processes of multiple frameworks, such as PMBOK and CMMI (Wangenheim, da Silva, Buglione, Scheidt, & Prikladnicki, 2010), ISO/IEC 12207:08, ISO 15504:5, 12207:08 and CMMI-Dev (Pino, Baldassarre, Piattini, & Visaggio, 2010), and, ITIL and ISO/IEC 15504 (Barafort et al., 2002).

Furthermore, there seems to be a common agreement among academics and researchers that existing SPI international standards are too complicated and expensive to be implemented by SSEs with up to 50 employees (Alexandre, Renault, & Habra, 2006; C Laporte, Alain, & Renault, 2006; Niazi, Babar, & Katugampola, 2008).

Thus, the success of SPI initiatives may only be possible for large companies that are able to employ enough resources to handle the initiative.

Although research is reporting that the software engineering community has shown an increasing interest in implementing SPI initiatives in SSEs (Pino, García, & Piattini, 2008), yet for many modern SSEs using international standards to manage their software development activity has been a major challenge. Additionally, this challenge will increase in complexity when SSEs are forced to adopt multiple standards. For this reason, prior research has reported a number of initiatives led by governments to guide SPI programmes in SSEs (A Cater-Steel, Toleman, & Rout, 2006; Pino et al., 2008); yet, evidence reveals that a small number of SSEs have been reported to implement SPI (Lester, Wilkie, McFall, & Ware, 2010; F Pino, M Baldassarre, et al., 2010).

1.6 Why Small Enterprises?

Small enterprises comprise the largest population of the world software industry (A Cater-Steel, 2001; C Laporte et al., 2006). According to the New Zealand Ministry of Development (2010), 97% of enterprises in New Zealand employ 19 or fewer employees, and small medium enterprises (SMEs) dominate most industries. As reported by (A Cater-Steel, 2001) the Australian software industry is one of the fastest growing industries in Australia and is dominated by small and very small businesses. Software teams of fewer than 50 employees constitute the larger proportion of the industry in Europe, Canada, Brazil and the rest of the world (Pino et al., 2008) . The literature review on SEs and SSEs, as will be shown in chapter 2, highlights the similarities between them. The review shows that SSEs have varying capability levels in their technical and organisational processes and wide variation in the extent of adoption of best practice (A Cater-Steel, 2004b).

1.7 Distinctiveness and Significance of this Study

The absence of a holistic multifaceted approach to process improvement programmes in software organisations has been a barrier to the success of such endeavours in SSEs (Kirk & MacDonell, 2009). As the literature review (see Chapter 2) will show, there is a common understanding among researchers from different disciplines about the problems faced by SSEs. However, in their efforts to find a solution to the problem, researchers from each research field have taken a different approach with a result similar to that of the blind men in the elephant parable. Their proposed solutions depended entirely on research published in their field of specialisation. Hence, the revealed truth is limited due to the failure to account for other truths or a totality of truth. This gap in SPI approaches has made it difficult to offer frameworks that can guide managers to improve the maturity of practices in their firms. To overcome this problem, this study takes a holistic approach by including multiple perspectives to ensure that all aspects of the problem are considered. This holistic view is one of the objectives of this research project and is achieved by taking a systematic approach that includes a number of steps, as follows.

Firstly, a comprehensive literature review of the research in the area of Small Organisations in the business management field was conducted. This step provided insights on the issues, success and failure factors shared by small-size organisations and unique issues faced by SSEs.

Secondly, a comprehensive review of SPI literature was undertaken to identify the challenges that would help or hinder the adoption and improvement of processes in SSEs.

The software process reference capability model is the main outcome of this research study. The proposed model adopts multiple international standards for different areas of the business to provide SSEs with a more holistic approach, as recommended by Barafort et al. (2002), who argued that the adoption of standards from multiple disciplines provides a blend of talents that can address multi-faceted problems.

Finally, a review of strategic management literature related to Dynamic Capabilities was carried out, with the objective of enhancing the proposed model. In this context, processes are not only created to structure the organisation, but also to be a source of competitive advantage to enable managers of software development organisations to sense, seize and transform opportunities and create new sources of competitive advantage. This view has led to the identification of a number of dynamic capabilities practices in the model.

Therefore, the proposed capabilities and underlying processes constitute a holistic multi-faceted capability process model that can bring value to all stakeholders and provide a common language and approach to the development of organisational processes in SSEs, as well as a more adaptive approach to SPI programmes in organisations. Hence, this study contributes to the body of knowledge on Dynamic Capabilities, SPI and Small Organisation.

1.8 Research Scope

This research explores the application of SPI practices in SSEs, whether for establishing effective organisational processes or enhancing the enterprise's maturity level. In terms of size, the focus will be on SSEs with 50 employees or fewer in addition to IT departments in medium and large size enterprises with very small structures that are engaged in the provision of software solutions.

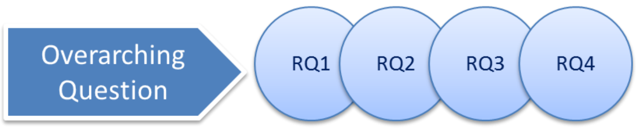
1.9 Research Aims

The ultimate aim of this study is to propose a process reference model that will guide small software development organisations in managing successful software development projects and process improvement initiatives. This proposed model will seek to: 1) provide the processes needed to help organisations meet their perceived project outcomes, 2) be informed by dynamic capability research, 3) build upon existing software engineering standards, 4) provide a multi-faceted approach without the need to implement multiple SPI standards, and 5) be feasible for application in SSEs.

The model intends to address several organisational elements that influence the quality of the developed software, including organisational, technical, project management and process improvement by adopting a holistic approach to the development of software.

1.10 Research Questions

Higgs, Horsfall and Grace (2009) suggest that it is good practice to start with an overarching research question that frames the research project and provide sub-questions to make the research more focused and manageable. Therefore, to achieve the research aim, the investigation is primarily concerned with the following question:



Overarching Question

What do experts in software development projects construe as relevant practices and Dynamic Capabilities that make up a holistic multi-facet software process improvement model?

Sub-Questions

1. What is reported by prior research about the issues faced by SSEs in their implementation of SPI initiatives?

2. What are the specific capabilities critical for the desired model that would benefit SSEs at each phase of the software product lifecycle?
3. What are the underlying processes desired for each identified capability suitable for SSEs?
4. What are the potential dynamic capabilities suitable for SSEs?

Table 1-1 outlines the research method and how the research questions were addressed.

1.11 **Research Approach**

The study will employ a post-positivist approach and a Delphi technique. Quantitative and qualitative responses will be obtained concurrently using 2-3 rounds questionnaires. The quantitative data will be analysed to assess the consensus among panellists on the importance and feasibility of the proposed practices.

The qualitative data will be analysed to corroborate and validate the quantitative results and to gain more insights on the model.

The process will involve analysing collected data, assessing group judgement or views, and facilitating the interaction between panel members through a controlled feedback to help them to achieve a consensus on the final model. Panellist will be asked to recommend changes by adding or removing practices, and by suggesting changes to the model to make it workable for SSEs.

1.12 **Addressing the Research Questions**

The following diagram and table illustrates how the research questions map to the research objectives and methods.

Table 1-1 : Addressing Research Questions

Research Aims	Research Questions	Research Objectives	Method
What do experts in software development projects construe	**RQ1.** What is reported by prior research about the issues facing	• To identify SEs' characteristics and challenges, which contribute to their success or failure,	➢ Detailed review of SE literature ➢ Relevant statistics on

Research Aims	Research Questions	Research Objectives	Method
as relevant practices and Dynamic Capabilities that compose a holistic multi-facet software process improvement model? The proposed model intends to: 1) provide the processes needed to help organisations meet the perceived project outcomes, 2) be informed by dynamic capability research, 3) build upon existing software engineering standards, 4) provide a multi-faceted approach without the need to implement multiple SPI standards, and 5) be feasible for application in SSEs.	SSEs in their implementation of SPI initiatives?	based on the research conducted in this field and statistics retrieved from trusted sources. • To identify SSEs' characteristics and challenges that would help or hinder the adoption of processes and the success of SPI initiatives based on research conducted in this field.	small firms
	RQ2. What are the specific capabilities critical for the desired model that would benefit SSEs at each phase of the software product lifecycle?	• To identify a set of capabilities that enable SSEs to reach the necessary level of maturity to meet the perceived project outcomes based on relevant research conducted in this area. • To seek the opinion of field experts on the proposed capabilities.	➢ Development of a capability model. ➢ Delphi Study
	RQ3. What are the underlying processes desired for each identified capability suitable for SSEs?	• To compare the latest versions of several popular multi-disciplinary standards; these are used collectively to improve organizations' levels of maturity. • To identify efficient processes from each standard that are considered core to SSEs. • To identify effective processes for each identified capability. • To develop a model that incorporates multiple standards	➢ Comparison of relevant SPI standards. ➢ Delphi Study

Research Aims	Research Questions	Research Objectives	Method
		from multiple disciplines. • To seek the help of field experts to improve the proposed processes and sub processes for each capability.	
	RQ4. What are potential dynamic capabilities suitable for SSEs?	• To identify Dynamic Capabilities to enhance the proposed model and to enable them to survive in a dynamic environment based on research conducted in this field. • To identify a framework for incorporating DCs into the model. The sense and respond framework has been selected. • To seek the help of field experts to improve the proposed model by incorporating DCs into the model.	➢ Review of DC literature. ➢ Delphi Study

1.13 Outline of the Thesis

CHAPTER 1 – INTRODUCTION: The objective of this chapter is two-fold; first, it communicates the motivation, aims and objectives of the research study. Second, it identifies the scope, the gap in literature and the contribution of the study to understanding the phenomena and the problem. Chapter 1 lays the foundation for the literature review in the next chapter.

CHAPTER 2 – LITERATURE REVIEW: This chapter reviews the relevant statistics and literature of Small-Size Organisations, SPI and Dynamic Capabilities in detail. Moreover, it presents a short overview of existing SPI standards that were included in the development of the model. This chapter provides a foundation for designing the research methodology and conducting the Delphi method.

Chapter 3 – Capability Model: This chapter discusses in detail the approach followed in defining the five capabilities that form the structure of the proposed model and provides justifications. It includes a short description of how the comparison of international standards was carried out; this work leads to the identification of the processes and sub-processes.

Chapter 4 –Methodology: This chapter discusses and provides justifications for the research philosophy and the Delphi method used to collect data. This chapter forms the basis for Chapters 5 and 6.

Chapter 5 – Analysis and Findings: This chapter presents the analysis and findings of the Delphi study and how the research objectives were met and questions answered. This chapter is an important prerequisite for Chapter 6.

Chapter 6 – Discussion: States the interpretations of the research findings, and how they correlate to the literature review in Chapter 2.

Chapter 7– Conclusion: This chapter presents the conclusion of the research and addresses the limitations and potential future research in this area.

The V-Model presented below display how the different chapters link together and shows the connection between intention and outcome.

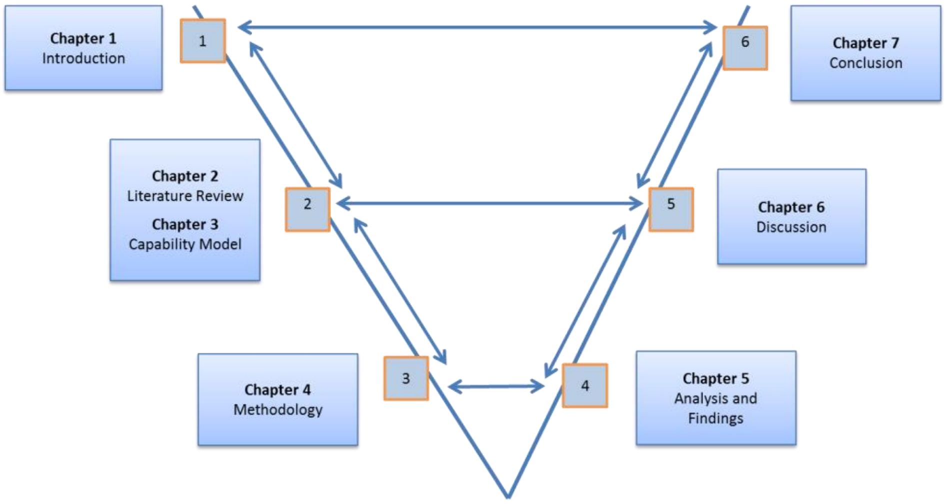


Figure 1-1: Model presenting the layout of the Study

CHAPTER 2 - LITERATURE REVIEW

2.1 Introduction

This chapter reviews the state of the current research as it relates to the study objectives stated in Chapter 1. As demonstrated in Figure 2-1 below, the chapter attempts to answer the research question: *RQ1 What is reported by prior research about the issues facing SSEs in their implementation of SPI initiatives?* Moreover, it contributes to answering *RQ4 What are potential dynamic capabilities suitable for SSEs?*

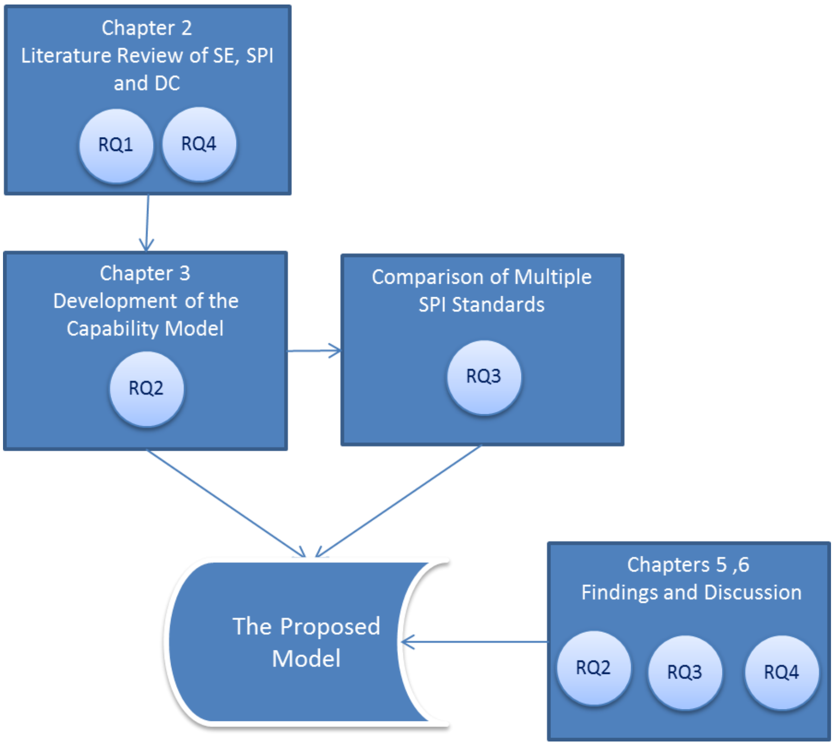


Figure 2-1: How the research method addressed the questions

2.2 The Need for Holistic Approach

Software development is a complex endeavour, which has to deal with matters of varying nature such as applying project management skills, managing the human aspect being the stakeholders, managing the system architecture artefacts and managing the economic aspects of the software development activities. The introduction chapter of SWEBOK (2004) emphasises a strong relationship between the software engineering discipline and eight related disciplines including computer science, computer engineering, management, mathematics, project management, quality management, software ergonomics and system engineering. Zavala-Ruiz (2008) holds this view and affirms that software engineering is not solely an engineering discipline, it also draws on business management and is social in nature.

This complex nature of software engineering encouraged researchers to examine different approaches to improve the software development process. Much of the efforts followed an engineering approach focusing on introducing new tools, methodologies and improving processes (Patnayakuni & Ruppel, 2010). Some research efforts acknowledged the social nature of the process, and hence adopted a different approach. To give an example, Patnayakuni and Ruppel (2010) adopted the socio-technical view suggesting that by improving the tools, methodologies and processes together with work design will only result in a more positive outcomes (Patnayakuni & Ruppel, 2010). Meanwhile, there have been substantial efforts to improve the quality of the software product by focusing on upgrading the organisation's processes through assessment and analysis against international standards, such as the international standard *Capability Maturity Model Integration* (CMMI) or other international standards with similar purpose. Such approaches are referred to as Software Process Improvement (SPI) programs.

Therefore, the above short discussion underscores that whatever approach researchers select to study the development of software in an organisation, there is a need for a holistic approach that focuses on the entire organisation system rather than the technical engineering aspect or whatever seems to be broken. This view is supported by Kirk and MacDonell (2009) who stressed the need for SPI approaches that focus on the whole organisation system besides consideration of cost and quality.

In light of the above discussion, this study attempts to take an analytical stance seeking a holistic view to address the issues and matters influencing the development of software projects in small enterprises. Therefore, it has adopted a multidisciplinary approach by reviewing and analysing the literature from multiple areas precisely the software engineering, small organisation management and the strategic management area of dynamic capability.

Subsequently, the study has considered what the introduction chapter highlighted, that the current international standards are too complex for SSEs. Therefore, in this research effort, special attention has been given to the specific characteristics of small enterprises (SEs) in general and SSEs in particular and the underlying issues facing them.

Further, it considered the highlighted limitation of existing SPI programs that existing efforts have been strongly focusing on the technical aspects of the organisation. Additionally, the study introduces the dimension of dynamic capabilities approach, which opens new ways of thinking about strategic planning and enterprise survival in dynamic markets, as it has been reported by research that SSEs are very weak with strategic planning. Finally, the study will analyse the practices of the most prominent SPI standards in an attempt to introduce an effective multifaceted process capability model based on accepted practice to provide SSEs with the needed processes to run successful business and for successful software project outcomes.

To address the above underlying study directions, this chapter is divided into three parts. Part I provides a review of literature on the characteristics and issues common to all SEs in general and further reveals the characteristics and challenges of SSEs in particular. Part II provides a discussion on the background of SPI programmes and their importance to businesses involved in software development and discusses factors that can inhibit or enable the adoption and improvement of processes in SSEs. Finally, Part III offers a review on dynamic capabilities theory and practice and applies this theory to the model to enhance the organisation adaptability and survival in dynamic markets.

Figure 2-2 presents what the literature review attempts to achieve.

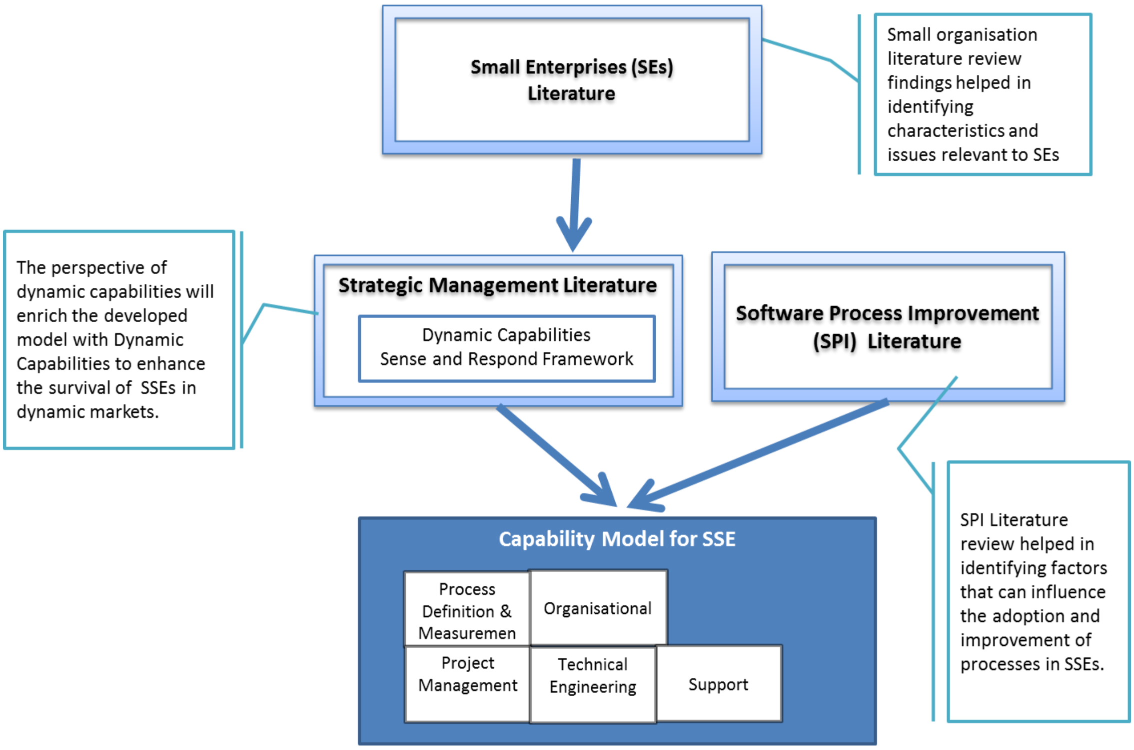


Figure 2-2: Literature Review Process

2.3 Part I - Review of Small Organisation Management Literature

2.3.1 Defining Small Enterprises

Within the current literature, there is no single, uniformly acceptable definition of a small business. The terms firm, organisation, enterprise, small settings and start-ups have been used by different writers, however they are treated interchangeably here.

Most of the research articles refer to employee count as a determinant factor for enterprise size. Employee count is the head count of salary paid to employees, sourced from taxation data "SMEs in New Zealand: Structure and Dynamics 2011" 2011). A uniform appropriate size of small enterprises is not clearly defined and can differ across countries and published statistical reports. The European Union (EU) defines micro enterprises as having fewer than 10 employees; small enterprises having at least 10 but fewer than 50 employees. The US defines small enterprises as having fewer than 500 employees.

While this study acknowledges that using employee count as a measure of firm size may cause inconsistencies, however this measure is the most commonly used categorisation. This study adopts the OECD (2010) definition of small enterprises provided hereunder which is being used by the main OECD countries including New Zealand and Australia.

> An enterprise is a legal entity possessing the right to conduct business on its own; It may consist of one or more establishments situated in a geographically separate area. Small enterprises refer to those with fewer than 50 employees.

Since the scope of the study is not limited to New Zealand, small enterprise is defined as any type of business with fewer than 50 employees. In New Zealand small represents business with (6 to 19 employees), small to medium (20 to 49 employees) ("SMEs in New Zealand: Structure and Dynamics 2011," 2011, p. 10).

2.3.2 Small Enterprises as Economy Key Players

Most of the world is dominated by small firms, which make significant contribution to the economy in terms of the number of enterprises and the proportion of the labour force employed by these firms.

The number of enterprises with less than 20 employees exceeds 70% in many countries, ranging from 67% in Ireland to 95% in Greece. The contribution to employment ranges between 11% in the United States and Czech Republic and more than 35% in Greece (OECD, 2010).

As presented in Table 2-1, New Zealand’s enterprises are predominantly small size, with 97.2% of all private and government enterprises employ 19 people or below at February 2010 ("SMEs in New Zealand: Structure and Dynamics 2011," 2011). As illustrated by Table 2-2, SMEs make a significant contribution to employment in New Zealand, accounting for 30.7 percent of total employment at February 2010. As shown in Figure 2-3, SMEs’ contribution to New Zealand total value added of gross domestic product (GDP) in 2009 was 39.8 percent.

Table 2-1: Enterprises by size ("SMEs in New Zealand: Structure and Dynamics 2011," 2011)

Employee Size Group	Number of enterprises	Percentage of all enterprises	Cumulative percentage
0	323,935	68.9%	68.9%
1-5	97,888	20.8%	89.7%
6-9	19,571	4.2%	93.8%
10-19	15,980	3.4%	97.2%
20-49	8,420	1.8%	99.0%
50-99	2,489	0.5%	99.6%
100-499	1,739	0.4%	99.9%
500+	324	0.1%	100.0%
Total	470,346	100%	-

Table 2-2: Employment by SMEs ("SMEs in New Zealand: Structure and Dynamics 2011," 2011)

SMEs	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
Total SME employment	503,240	523,990	545,510	563,530	583,180	591,400	598,880	606,970	590,560	580,680
Percentage change from previous year		4.1%	4.1%	3.3%	3.5%	1.4%	1.3%	1.4%	-2.7%	-1.7%
As a percentage of total employment	31.1%	31.5%	31.7%	31.4%	31.4%	31.3%	31.1%	30.7%	30.6%	30.7%

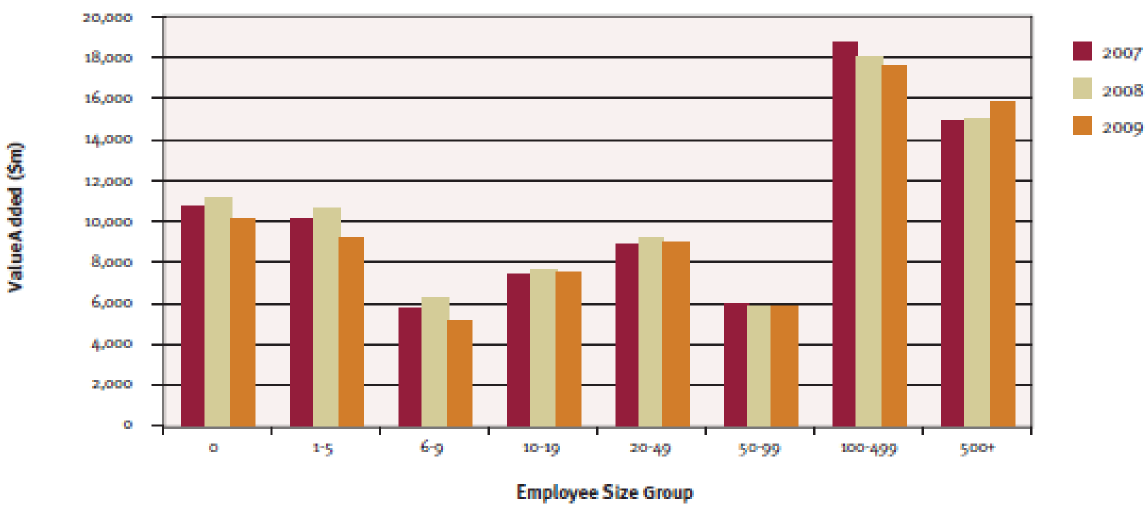


Figure 2-3 Total Value-Added by Employee Size Group ("SMEs in New Zealand: Structure and Dynamics 2011," 2011)

The software industry in most countries is mainly composed of small enterprises, which also contribute to the development of national economies. As reported by C Laporte et al. (2006): 'in Europe 85% of the Information Technology (IT) sector have between 1 and 10 employees. In the Montreal area, 80% of IT companies have between 1 and 25 employees, in Brazil, small IT companies represent about 70% of the total number of companies and finally, in Northern Ireland 66% of IT organizations employ fewer than 20 employees.

Table A-1 in Appendix A provides a summary of statistical reports findings.

2.3.3 Characteristics of Small Enterprises

From the review of many articles, numerous advantages and challenges differentiate small enterprise (SE) from large enterprise (LE). The structure, management and processes of SE are different from the professional management style practiced in LE, which are practiced on a reduced scale. As reported by Zavala-Ruiz (2008) there is significant evidence that the economic performance of SEs is not inferior to LEs.

Woods and Joyce (2003) described SEs as having limited human resources, loose division of labour, minimal differentiation among its business units, flat structure and non-formalised administrative behaviour characterised with minimal planning and staff training.

SEs are normally focused on the day to day activities and are less likely to utilize strategic management models and long term planning (Pryor, Toombs, Anderson, & White, 2010). The lack of strategic planning processes has been linked to the poor performance of SEs (Fening, Pesakovic, & Amaria, 2008). Although the aim of most SEs is to grow, many of the managers lack the background and expertise to manage the business strategically. Attention has been given by researchers to the benefits of strategic planning (Mazzarol et al., 2009).

Much research has also focused on factors influencing SE growth and performance. The role of owner/manager on the performance and growth of business is highly emphasised by the literature (Mazzarol et al., 2009). The unique and multiple roles played by the owner-manager degrade the decision-making process and lead to inappropriate actions (Zavala-Ruiz, 2008).

Attempts have been made to identify the behaviour and attributes associated with the enterprise people. There is evidence that rapidly growing firms are linked to the entrepreneurial orientation of the organisation or entrepreneurial drive of the individuals in the organisation (Wood & Michalisin, 2010). Barkham, Gudgin, and Hanvey (1996) found that the entrepreneurial characteristics of managers and the adopted business strategies determine the growth of small firms.

Despite all the negative characteristics, the size of small enterprises allows them to be more adaptable and responsive to changing market conditions than larger enterprises as it is easier for one decision maker to manage the business effectively, react to market changes, and exploit new opportunities(Abdelrahim & MBA, 2007). These characteristics allow them to grow faster than LEs.

Due to these unique characteristics a number of research studies emphasised the need for efficient practices adapted to their size and maturity level (C Laporte et al., 2006; Pino et al., 2008).

A summary of the literature review findings on the characteristics of SEs is presented in Table A-2 in Appendix A.

2.3.4 Small Enterprises Success and Failure Factors

Various studies highlight the high level of bankruptcy and failure among SE and advise that an understanding of the factors or practices that lead to the firm's success or failure is crucial to the stability of the economy (Lussier & Halabi, 2010; Lussier & Pfeifer, 2000).

Empirical studies exist which have introduced a set of key factors or practices that can differentiate between successful and failing SEs. These factors are referred to as success and failure prediction models.

The most widely accepted study was the Lussier (1995) that provides a non-financial success and failure prediction model that included 15 factors empirically tested for their effectiveness. This model was originally tested using a sample from United States, then Croatia and last from Chile (Lussier & Halabi, 2010) and was also tested on different types of industries (Lussier & Pfeifer, 2000).

The main factors that was found to significantly contribute to SE's success were education and experience of management and staff, the existence of detailed plans, and the use of professional advice (Lussier & Pfeifer, 2000). Table 2-3 provides a list of the 15 independent successes versus failure factors identified in the model.

Table 2-3: Lussier (1995) success versus failure variables (extended model)

Capital: Businesses that start undercapitalized have a greater chance of failure than firms that start with adequate capital.
Record keeping and financial control: Businesses that do not keep updated and accurate records and do not use adequate financial controls have a greater chance of failure than firms that do.
Industry Experience: Businesses managed by people without prior industry experience have a greater chance of failure than firms managed by people with prior industry experience.
Management Experience: Businesses managed by people without prior management experience have a greater chance of failure than firms that are managed by people with prior management experience.
Planning: Businesses that do not develop specific business plans have a greater chance of failure than firms that do.
Professional Advisors: Businesses that do not use professional advisors have a greater chance of failure than firms using professional advisors.
Education: People without any college education who start a business have a greater chance of failing than people with one or more years of college education.
Staffing: Businesses that cannot attract and retain quality employees have a greater chance of failure than firms that can.
Product/ServiceTiming: Businesses that select products/services that are too new or too old have a greater chance of failure than firms that select products/services that are in the growth stage.
Economic Timing: Businesses that start during a recession have a greater chance to fail than firms that start during expansion periods.
Age: Younger people who start a business have a greater chance to fail than older people starting a business.
Partners: A business started by one person has a greater chance of failure than a firm started by more than one person.
Parents: Business owners whose parents did not own a business have a greater chance of failure than owners whose parents did own a business.
Minority: Minorities have a greater chance of failure than non-minorities.
Marketing: Business owners without marketing skills have a greater chance of failure than owners with marketing skills.

The success versus failure prediction model is only one tool for understanding SE behaviour, however, it is an important one as it helps in understanding why firms fail and succeed. It also guides managers of SSEs to redifine their practices by learning from the model findings.

2.3.5 Characteristics of Small Software Enterprises

A characterisation of Small Software Enterprises (SSEs) has been reported by researchers based on their analysis and observation of process improvement initiatives in their organisations. The reported characteristics of SSEs do not differ much from what is reported by literature on the management of small enterprises as illustrated in the literature review above. (A Cater-Steel et al., 2006) highlighted the participation and involvement of owner-managers in all aspects of the business and SPI activities.

According to Zavala-Ruiz (2008) most of the productive, administrative and management processes of SSE continue to develop in a chaotic manner and the software products are delivered in a culture of urgency and completion. People are continuously focused on meeting the tight deadlines with minimum attention given to the quality of the final product (Alexandre et al., 2006).

The limited resources of SSE enterprises is emphasised by research (Alexandre et al., 2006; Mishra & Mishra, 2007). The lack of resources assigned to the software process exists at all levels including monetary, human, time and in particular, the quality processes suffer the most. (Alexandre et al., 2006; Pino, Pedreira, Garcia, Rodriguez, & Piattini, 2010; Zavala-Ruiz, 2008). Due to this, the maturity of the software process normally ranks very low and subsequently business focus is primarily on project and technical matters than organisational matters (A Cater-Steel, 2001).

F Pino, O Pedreira, et al. (2010) point out different characteristics, including having a flat structure with free management style and lightweight processes with constant communication between project members and customers.

Carter-Steel A. (2001) reported a number of factors that hindered the success of SPI efforts in the four firms she had assessed. Among these factors is the shortage in IT staff and resources, and when key staff are lost it becomes difficult to find replacements or send existing staff to attend appropriate training.

Alexandre et al. (2006) the developers of the OWPL, a gradual SPI framework designed for small business, reported a number of common characteristics, which can be generalised, on SEs.

- SSEs show remarkable competence in the technical domain but suffer high variance in their software and quality processes.
- The software lifecycle model is very simplified. Development and testing are the primary phases, although testing can be shortened to meet deadlines or to solve the issues due to the lack of human resources.
- Processes exist with varying quality levels; some practices can be very mature while others are extremely poor.
- Customer-supplier relationship practices enjoy high quality levels because they are legally and contractually imposed, but also are considered a burden rather than an asset by the organisation.
- Control procedures are very poor which affect the adoption of employees to new procedures. This is considered the basic cause behind the uneven quality in software processes.
- Project management and planning practices cannot be generalised across SSEs. Some follow good project management practices while others do not.
- Training and professional development practices are very limited due to the severe budget constraint.
- Lack of strategic planning.
- Learning and knowledge management is rarely observed.
- Because of their size, they cannot impose their methodological approach even if they are sound and they tend to follow the methodological guidelines of their larger customer.

- Flatter structure with informal communication. This can be seen as an advantage as it expedites decision making. However, a minimum level of formalisation can benefit the organisation.

A summary of the characteristics of SSEs is presented in Table A-3 in Appendix A.

2.3.6 Findings of Small Enterprise Literature

The small enterprises literature discussed above reveals that small enterprises face several challenges. It also highlighted similarities between the characteristics of SEs and SSEs; a comparison is provided in

Table A-4 in Appendix A. In this study, it is advocated that the factors that can inhibit or enable the performance of small organisations will have an impact on the success or failure of SPI initiatives.

The most critical challenge is the absence of strategic analysis and planning the positive impact of such practices on the firm's performance is strongly supported by empirical evidence. The literature review supports the need for a team of entrepreneurs in upper management to lead the strategic direction and strategic choices of the enterprise.

The literature review highlighted the critical role of the owner-manager to the performance of the business. The literature review findings show that the inability of the firms to take sound strategic decisions due to the interference of owner manager who is evident to impair the firm's entrepreneurial drive and accordingly the chances of development and growth. In addition, the managerial challenges facing the owner-manager increase as the enterprise progress into more mature life stages. Further, the need for coherent, skilled and a high entrepreneurially driven top management team is emphasised in the findings of the literature review on small enterprises.

Similarly, the literature review on SSEs shows similar findings such as the absence of formal and institutionalised practices mainly strategic planning and knowledge management.

Software development is an intelligent type of work highly dependent on knowledge workers; hence, the literature review suggests recruiting well skilled professionals is a key organisational success factor. This also emphasises the need for SSEs to harness and focus on the capability of employees by creating a learning environment. SSEs are characterised as having high employee turnover, and high number of part time contracts, which can disrupt the stability of the work environment and may have a negative impact on productivity and work efficiency.

These findings demonstrate the need for redefining management practices for SSEs, and calls for devoting more resources into strategic planning, professional management and the development of employees. Hence, the above findings collectively call for a business model that focuses software process improvement (SPI) processes and capabilities that particularly challenge the traditional ways to strategy, leadership, and the management of knowledge and human resources. The model should allow the firm to survive when time-to-market is critical, the rate of technological change is rapid, and competition and markets are difficult to determine (Teece, Pisano, et al., 1997).

2.4 Part II - Review of Software Process Improvement Literature

Software process improvement is a set of applied practices rooted in multiple disciplines predominantly software engineering, project management and management. Both industry practice and research have resulted in a variety of best practice normative standards and models for improving the organisation and the management of software projects. The practices dealt with software development issues at organisational, project and engineering levels. SPI initiatives include four major areas of focus: SPI assessment, process definition, institutionalisation and deployment of processes

SPI stakeholders comprise of a wide range of parties including government bodies, software consumer organisations, software supplier organisations, software engineers, academic bodies including institutes and researchers.

The literature points out the need for software development organisations for guidance to understand the practices proposed by different SPI standards especially small enterprises (A Cater-Steel, 2001). SSEs generally do not have the knowledge of the models and do not know how to define processes (Niazi, 2009). Additionally, SSEs require help on how to implement SPI initiatives (C Laporte et al., 2006) .

Although there are many SPI standards that share common features, the field is highly dominated at the very early stages by the Capability Maturity Model (CMM) from the Software Engineering Institute (SEI), which has influenced both the literature and practice (Hansen, Rose, & Tjornehoj, 2004).

Despite the numerous success stories, there is a high rate of variability reported on the benefits of SPI initiatives. Moreover, the majority of success stories represent large organisations. Some qualitative and quantitative evidence of success exist, but it is difficult to find consistent evidence to generalize to theory or present the benefits in financial terms (Galinac, 2009; Hansen et al., 2004). The SPI research has been continually challenged and criticised as lacking theoretical basis and overly prescriptive lacking reflective inputs (Hansen et al., 2004).

2.4.1 SPI Criticism

Reflective studies criticised SPI approaches as being rooted in Taylor's scientific management which assigns the responsibility of thinking and taking decisions to managers and the responsibility for implementing to workers and require work processes that must be scientifically established; it looks at enforcing command and control through processes set by management (Nielsen & Kautz, 2008).

Hansen et al. (2004) provided an unusual discussion that uncovered a number of management and organisational assumptions underlying the CMM standard. Those assumptions if accepted can be generalised to other SPI standards.

First, SPI is criticised as lacking business and organisational context. The primary philosophy of SPI approaches is rooted on the assumption that when the process is good the eventual product will be as good. Supporters of this school of thought believe that the models may prove successful in one organisational context but fail in the other due to the differing internal and external organisational circumstances. This view agrees with the findings of Cattaneo et al. (2001) in their report of their experience in conducting SPI initiatives in three companies, in which they concluded that effective process assessment and improvement models must pursue the coherence improvement of all aspect of the organisation not just of the engineering factors.

Second, SPI success requires top management careful observation and control to ensure that the employees are fully compliant with the processes. This will discourage personal initiatives and ownership of tasks since managers dictate the plans, activities, and details.

Third, standardisation reduces the ability of an organisation to adapt to change. It takes a long time to establish organisational processes and procedures and requires more time to enforce them organisation wide. Consequently, it requires an equally long time to change them. Supporters of this assumption believe that the lack of process formalisation does not imply poor processes and encourages the establishment of processes through less formal way.

Fourth, SPI initiatives, which are led by external consultants, fail to adjust to the internal and external organisational conditions and on-going environmental challenges.

2.4.2 Enablers of SPI Initiatives

Empirical studies based on the analysis of surveys and the findings of industry case studies report several factors as enablers of SPI.

Having SPI objectives tied to business objectives is identified as a critical factor to the success of SPI initiatives (A Cater-Steel, 2001).

One primary enabler factor identified in the literature is senior management commitment, which has been described as the bottleneck for the success of SPI initiatives (A Cater-Steel et al., 2006; Galinac, 2009; Niazi, 2009). Management commitment is crucial by providing human and other needed resources and including the assessment process as part of the employee workload. Employee compensations for SPI efforts as part of normal work assignments are influential factors in driving successful results (Goldenson & Herbsleb, 1995). Management commitment can be also manifested by ensuring the progress is monitored so that the agreed recommendations are implemented.

Managers actively monitoring the progress of SPI projects are more common in organisations that showed successful SPI results (Goldenson & Herbsleb, 1995). According to A Cater-Steel (2001) two thirds of SPI programmes in the late 1980 fail due to lack of commitment and follow-up in measuring the targeted improvements. The literature pointed out the importance to define scope to an achievable extent and the need to implement the action plans within a short period not exceeding 3-5 month timeframe SPI (A Cater-Steel, 2001).

2.4.3 Reported Limitations of SPI Initiatives

Drawing on empirical SPI literature, case studies and online articles, there are various implementation issues facing the SPI initiatives.

The lack of organisation's internal and external coherence is clearly suggested by researchers (Cattaneo et al., 2001; Dalcher, 2009), which hinder SPI projects from delivering successful results. In a three year study of three companies operating in the banking and financial market, Cattaneo et al. (2001) concluded that organisations focusing on the technical and engineering processes alone may not enjoy fruitful SPI outcomes. They stress the success of SPI requires them having internal coherence in which the engineering and technical capabilities are fine-tuned with other internal aspects such as strategy, human resources and culture. In addition, the organisation must be responsive to the needs of customers and the changing external environment and market place. Rousseva (2007) also underscores organisational capabilities and considers them critical drivers shaping the dynamics of the development of technological capabilities.

Although organisations have started to comprehend the importance of SPI efforts and the benefits of adopting them, there has been slow acceptance of SPI initiatives which can be related to the fact that their advantages were not noticeable within short timeframe (A Cater-Steel, 2001) . It takes a very long time for results to generate which makes it hard to assess their effects on the organisation in the short term. Software improvement is a gradual and staged process; it is a planned, managed and controlled effort that must be treated as a project with its own plans resources and commitment (Cattaneo et al., 2001).

Unrealistic goals and high expectations of results also have been reported (Nasir, Ahmad, & Hassan, 2008). Low process priority and lack of resources have been reported (Niazi et al.). A number of online articles reports lack of having unified goals, undefined performance indicators (KPI) and process control measurement. Some researchers find it impossible to apply quantitative assessment because the baseline was not collected at the start of the programme (Cattaneo et al., 2001), and this is likely to be the case of most organisations running SPI programmes. A number of studies as reported by Nasir Nasir et al. (2008) report the lack of staff training.

Moreover, implementing new processes may become an objective in itself and an end rather a means which may shift the focus more on the processes themselves rather than the initial objective of the initiative which may be improving organisation performance or maximising efficiency.

Despite the reported limitations and the lack of sufficient quantitative evidence on the apparent benefits and positive contribution of process improvement initiatives (Cattaneo et al., 2001; Zazworka, Basili, & Shull, 2010), studies are reporting qualitative findings about the adoption of process improvement standards which show an ultimate improvement in the organisation's maturity level (Cattaneo et al., 2001).

The existing SPI standards have however matured over the last ten years. The current standards have tackled a number of the issues that were raised by researchers in the late-1990s and mid-2000s. The transformation of CMM into the current CMMI product is just one example of this response. Moreover, the introduction of CMMI in its three flavours (development, service and acquisition) is also further evidence. IS0/IEC 15504 current state reflects the issues that have been addressed by literature. This diverse body of work is very prescriptive and addresses the needed focus areas for all types of organisations. The areas include product development management, project management, supporting processes such as configuration management, risk management and decision analysis, and process definition and improvement areas. However, in this study it is argued that the majority of existing standards still lack focus on the strategic and organisational areas.

2.4.4 Software Process Improvement Standards

Software process standards have evolved naturally, as modern businesses became more dependent on technology and more in need for practices that would enhance the level of predictability about the product quality and project success. Moreover, IT departments are challenged to align their practices with business strategies by applying rigorous best practice structures and standards.

All SPI standards are normative models, rather than providing detailed implementations in terms of methods, procedures, content and tools; they only define what must done to achieve the desired outcomes ("ISO/IEC 12207:2008 ", 2008) The implementation details are left to the users of the standards to develop, whether being separate documents or packaged and combined in some fashion, or even integrated into the organisation’s Quality Management System. Although this can seen as a flexibility by giving the users the choice in the selection of processes that suit their organisation or project. Hence, organisations remain free to innovate and adopt the processes within these rough and broad structures.

There are five major classes of standards which software development organizations can adopt, a brief description of each is provided below.

I. Software Engineering Body of Knowledge

The software engineering body of knowledge (SWEBOK) is emerging as a baseline for the body of knowledge for the field of software engineering. It complements the software engineering body of knowledge that already exists in the published literature. It describes the generally accepted portion of the body of knowledge and organises it into knowledge areas. It treats each knowledge area as a separate chapter in the guide.

II. Assessment Models

Assessment models aim to assess the processes and practices within an organisation in comparison to the promoted practice in the adopted SPI standard/s. They assess the organisation's processes to determine the capability level of each process, and identify gaps in the organisations practices.

These models are arranged in process areas. A process area is a set of related practices in an area when implemented collectively and consistently, significant improvements are ensured in that area (Wangenheim et al., 2010).

They can support two improvement paths: staged and continuous. The staged representation enables the organisation to increase their maturity level by improving processes from different process areas corresponding to the targeted maturity level. This improvement path increases the maturity level of the organisation incrementally and in stages focusing each time on the higher level. The staged representation makes it possible to compare the quality of practices among organisations using maturity levels, but it is not flexible as it requires a sequence of improvements (Ragaisis, Peldzius, & Simena, 2011).

On the other hand, continuous representation enables organisations to improve the capability of a set of processes related to an individual process area or group of process areas selected against a targeted process capability profile. With continuous representation, it becomes difficult to compare between the maturity levels of different organisations. It also provides detailed assessment on the capability of the process (Peldzius & Ragaisis, 2011).

The choice of representation is merely an organisational preference. Literature is not decisive on the most suitable architecture especially for small organisations (Peldzius & Ragaisis, 2011). Furthermore, a set of processes covered by the process area or maturity level may not represent the true process profile of the organisation, as the organisation may include additional primary business processes that are not included in the standard (Ragaisis et al., 2011).

Modern software development practices are moving towards outsourcing software development products and cloud computing services, which makes acquirers of services increasingly demanding software development maturity from their vendors and proof of compliance with international standards; therefore, adopting staged representation can be of significant help for both suppliers and acquirers of services.

Two prominent assessment standards are dominating this area of SPI standards. The Software Engineering ("Capability Maturity Model Integration for Development (CMMI-DEV) Version 1.3," 2010) and International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) ("ISO/IEC 15504-5:2012," 2012).

CMMI and ISO/IEC 15504 are considered infeasible for adoption in SSEs due to the cost and time involved, which exceeds the limited resources that SSEs can afford (Lester et al., 2010; Santos et al., 2007). Therefore, a number of tailored proposals were developed for SSEs based on CMMI and ISO/IEC 15504 as reported by Pino, Pardo, García, and Piattini (2010, p. 1045)

III. Software Development Life Cycle Models

Lifecycle models or reference models allow software development organizations to define their processes. They contain process activities and tasks that are grouped together into process groups to be performed during the life cycle of the software or service such as acquisition, development, operation, maintenance and disposal of software product or service.

The most dominant models in this area are: ("ISO/IEC 12207:2008 ", 2008) , and ("ISO/IEC/IEEE 15288:2015," 2015).

IV. Project Management Body of Knowledge

The Project Management Institute (PMI) has developed an international project management standard, the Project Management Body of Knowledge (PMBOK) (Rose, 2013). This standard provides a framework for managing projects. PMBOK is the most used project management framework (Ehsan, Malik, Shabbir, Mirza, & Bhatti, 2010). As a body of knowledge, PMBOK describes the key competencies that project managers must develop. It also defines five process groups that progress from initiating to planning, executing, monitoring and controlling to closing. Each process group defines a set of activities that span the different project phases and different knowledge areas.

In addition to PMBOK, CMMI and ISO/IEC 15504 provide a collection of best practices for project management. The comparison of standards in chapter 4 of this study show that all three standards share overlapping processes but there are differences. There has been efforts for harmonising the processes between the models by researchers such as the work of Wangenheim et al. (2010) who compared the similarities and differences of the project management processes of PMBOK and CMMI at a high level and proposed a single set of practices.

V. Service Management Models

The IT Service Management (ITSM) was initially designed to support the infrastructure architecture by describing processes and functions for an effective management and delivery of IT services to support business goals. However, the standards evolved seeking a philosophy focusing more than just operational effectiveness, but providing greater strategic positioning and continual improvements (M. Marrone, & Kolbe, L. M., 2011). The key contributions in this area is business and IT alignment, customer supplier relation, service portfolio management and the introduction of a methodological approach to the management of IT services through a lifecycle model, which focuses on end to end services from design to operation (M. Marrone, & Kolbe, L. M., 2011).

The most prominent and highly adopted ITSM standard is the IT Infrastructure Library (ITIL), which is considered a de facto standard for implementing IT service management (Hochstein et al., 2005; M. Marrone, & Kolbe, L. M., 2011). It has been reported in the literature that the adoption of ITSM especially ITIL is increasing globally (M. Marrone, & Kolbe, L. M., 2011; M. Marrone, Gacenga, Cater-Steel, & Kolbe, 2014). A related ITSM international standard is adopted by organisations is ISO/IEC 20000 (M. Marrone et al., 2014, p. 870)

The appropriateness of existing ITSM for SSEs is assumed to be similar to all other IT standards, however, empirical evidence supporting this assumption is lacking (M. Marrone et al., 2014).

This study, as has been discussed earlier in the first chapter, will include ITIL as one of the selected standards used to build the proposed model. ITIL processes, both operational and strategic will be explored when building the proposed model (M. Marrone, & Kolbe, L. M., 2011).

2.4.5 Driving Forces for Adopting SPI Standards

A number of driving market forces are forcing small and large organisations to demonstrate compliance with industry standards.

I. Outsourcing Governance

The recent years have marked an increase in outsourcing of services and infrastructure to third parties as a means to reduce cost and acquire specialised expert knowledge. This includes moving to the cloud and outsourcing software development to service providers in low-cost countries such as India, China and others. This trend for outsourcing has exposed service providers to high competition; hence forced organisations to adopt standards to guide them in establishing measurement in the assessment of vendor's capabilities and level of maturity. In addition, successful outsourcing requires a mutual governance framework for both customers and service providers to manage the relationship beyond the terms of the service level agreements (SLAs) and contracts.

II. Maximise Chances of Project Success

The massive advancement in technology such as virtualization, outsourcing and cloud computing have continued to add to the complexity of software systems. As the complexity increases, the quality of the product becomes a major concern. Studies show negative correlation between project size and success with coding errors and programming defects being contributors to project failure (Miller, 2007).

Improvement in the software process will result in improving the activities undertaken by the development team. It increases process visibility and improves programmer capability (Mishra & Mishra, 2007) through the adoption of formal processes such as clear communications, thorough observations, robust designs, accurate risk assessment and controlled change to requirements and progressive monitoring and control measures which are seen to enhance the chances of project success.

III. The Quality of the Software Product

As the complexity of IT systems is increasing there is a need to adopt best practice methods which describe how successful organisations approach software development. This movement has led to the awareness that the quality of a system is influenced by the quality of the selected software development processes (Zazworka et al., 2010).

IV. Reduce the Need for Rework

Studies have revealed that the cost of fixing a defect within the development process has proved to be 10-100 times cheaper when removed during the operation of the system (Miller, 2007). Improving quality on this ground has put the foundation and presents the need to focus on the quality of the software process. Therefore, it has become an accepted premise that the quality of the product is improved as the result of continuously assessing and improving the process as reported by Cattaneo et al. (2001); Miller (2007); Zazworka et al. (2010). (Wiegers, 1993) even argues that the focus on high quality will increase team productivity as a result of reduced maintenance over the system lifetime.

2.4.6 Findings of SPI Literature Review

The review of SPI literature shows that the journey to achieve excellence in small software development enterprises is a challenging and complex one.

Based on the literature review of SPI, we expect SSEs to have varying capability levels of their technical and organisational processes and a lack of consistency in their practices across different projects.

The fact that SEs can implement selected SPI elements as effectively as large organisations and that the size of the organisation will not limit the success of SPI initiatives (Lester et al., 2010, p. 18), leads us to argue for the need for effective non-bureaucratic lightweight processes derived from international standards. With sound processes SSEs can produce high quality product and promote strategic and operational planning which is a success factor identified in the literature review. It can have significant reduction in inspection cost and rework time (Mishra & Mishra, 2007).

When adopting process-based practices, all processes are documented and employees go through a learning process that can be shared at project level and integrated into the organisation learning processes. Hence, if an experienced staff member leaves, the organisation can still survive and the system will not be affected. This highlights the need for practices that are more process oriented, as it is likely to stimulate an organisational learning environment.

Finally, the coherence of technical engineering processes with internal aspects such as strategy and culture, along with external aspects such as customers and the market place is an enabler to the success of SPI initiatives.

Therefore, this study adopts the view that the compliance with one or more SPI standards must not be looked at as an optional practice, regardless of the organisation size, as process-based practices stimulate organisational coherence (if properly implemented) and promote consistency in the delivery of product or service.

The model proposed by this study offers a practical solution to a number of issues identified in the SPI literature review. It provides guidance for applying process-based practices in SSEs. The practices in the model address different aspects of a software organisation not just the technical engineering practices. Moreover, the model adopts dynamic capability theory discussed in the following section, which equips the enterprise with the capabilities to respond to changes in the business environment and promotes coherence between its internal and external environments.

2.5 Part III - Review of Dynamic Capability Research

2.5.1 History and Theoretical Background

In a globally integrated market, new forms of technology and competition are constantly arising causing constant and discontinued changes in the business environment. Research shows that in highly competitive environments, enterprises find it harder to achieve long-term competitive advantage (Barreto, 2010). This implies that the enterprise must find ways to respond successfully to these sudden shifts in the business environment. Dynamic Capabilities is one class of strategic management approaches that emphasises organisational effectiveness and efficiency and helps the firm to adapt quickly to changes in the business environment.

Dynamic capabilities have emerged as a conceptual framework in the field of strategic management research over the past 20 years (Eisenhardt, 2000; Helfat et al., 2009; Teece, Pisano, et al., 1997; Zollo & Winter, 1999). The framework advocates that firm’s effectiveness come from the possession of operational and strategic processes by which the firm can change its resource base in face of competition and changes in the business environment. Especially, those processes that use resources and those that can yield new resource configuration as the business conditions change (Eisenhardt, 2000).

Dynamic capabilities as a strategic management approach is still in its infancy and has not been accepted yet as a fully mature strategic management theory (Barreto, 2010; Dangol, 2012). The work remains mostly focused on conceptual level issues (Helfat et al., 2009). Substantial anecdotal and empirical research has been published in top management journals as reported by Barreto (2010), intending to advance the framework.

The literature shows that DCs is a promising approach both in terms of future research and as an aid to managers highlighting newer sources of competitive advantage based on firm-specific factors.(Teece, Pisano, et al., 1997). However, the complexity of the topic and the theoretical underpinnings generated some confusion among researchers. In fact researchers in this field admit that DCs broad theoretical development, which was drawn from a variety of perspectives[1], is increasing the challenge for creating a coherent theory (Helfat & Peteraf, 2003) . This made research in this area take different forms depending on the theory they have drawn on. Earlier research explored the contribution of the theory to firm performance while work that is more recent focused on organisational processes. Those researchers take the view that dynamic capabilities do not necessarily improve firm performance but allow

[1] Dynamic capabilities framework is considered an expansion to the RBV (Helfat & Peteraf, 2003). However, this view is controversial (Dangol, 2012; Helfat et al., 2009) because DCs draws on theories and research areas other than RBV such as to the evolutionary theory, entrepreneurship and transaction cost economics theories (Dangol, 2012), behavioral theory, organizational growth, routines and processes, organizational learning and managerial decision-making (Helfat & Peteraf, 2003).

organisations to pursue more operational effectiveness and respond to exogenous change (Helfat et al., 2009). The field still requires further empirical research to resolve the existing conceptual issues (Barreto, 2010; Helfat & Peteraf, 2003).

However, it is worth mentioning that the variations in the theory among researchers are justifiable by the youth of the framework (Barreto, 2010). There is a common agreement among researchers on the core concepts underpinning this framework (Helfat & Peteraf, 2003), which can benefit modern businesses. Moreover, there is an accepted work of empirical evidence supporting DCs. Therefore, the inclusion of DCs in this study is expected to enrich the applicability of the model.

2.5.2 Dynamic Capabilities as a Strategic Management Approach

According to Teece, Pisano, & Shuen (1997) the fundamental objective of the field of strategic management is to provide explanations for how firms achieve competitive advantage. Cambridge Business English dictionary defines competitive advantage as:

> "*The conditions that make a business more successful than the businesses it is competing with or a particular thing that makes it more successful*".

Numerous theories spread over the years, stress the achievement of competitive advantage through the exploitation of market power by utilising either product and market positioning or interactions between rivals. The most prominent among all are Porter's competitive forces (Porter, 1979) and strategic conflict approaches (Teece, Pisano, et al., 1997). Researchers have voiced concern that such strategic approaches are dependent on managers' intellectual ability to play smartly the rivalry game with competition, which is likely to distract them from the development and protection of unique firm-specific skills and capabilities that are major sources of competitive advantage (Teece, Pisano, et al., 1997).

This concern has opened the doors wide for a different class of strategic approaches, which shifted the focus of the organisation from market power to emphasising organisational effectiveness and efficiency through firm-level strategies. Both the

Resource Based View (RBV) and Dynamic Capabilities (DCs) approaches fall under this class of approaches.

Dynamic Capabilities uses concepts adopted by the RBV; both are efficiency-based approaches for understanding how competitive advantage within firms is achieved through firm-level efficiency advantages (Eisenhardt, 2000; Priem & Butler, 2001; Teece, Pisano, et al., 1997). RBV hypothesizes that competitors differ in their resources and capabilities and that when a resource owned by a firm is valuable and rare, it can give the firm competitive advantage. In addition, when such a resource is hard to imitate by competitors and is not substitutable, it can be then the source of sustainable competitive advantage (Priem & Butler, 2001). In order to generate sustainable competitive advantage, a resource must be a source of profit, and must be owned or controlled by the firm (Eisenhardt, 2000; Priem & Butler, 2001; Teece, Pisano, et al., 1997). Such resources or capabilities are considered strategic; examples of which include organisational processes, information, and knowledge (Kivelä, 2007; Priem & Butler, 2001). Strategic resources and capabilities enable the firm to implement strategies that can improve its efficiency and effectiveness, exploit opportunities or reduce the effect of threats (Priem & Butler, 2001).

The RBV theory applies only to static environments. It has not explained how to sustain competitive advantage in conditions of rapid changing business environment (Eisenhardt, 2000). The Dynamic Capabilities view, by contrast addresses the positive relationship between firm performance and its ability to cope with rapid change in technology and market forces (Teece, Pisano, et al., 1997). DCs framework requires dynamic interaction with the business environment to realign the firm's resource base with it to gain wealth. Hence, DCs look at continuously integrate, reconfigure and renew the firm's resource base and the utilisation of existing internal and external capabilities and the development of new ones.

2.5.3 What are Dynamic Capabilities?

The traditional competitive sources of differentiation that were based on economies of scale are no more worthwhile option for organisations. Hence, the accumulation of large stock of valuable assets is not enough to provide significant competitive advantage (Teece, Pisano, et al., 1997). The new basis of competitive advantage requires the enterprise to own, control or have access to capabilities in order to provide rapid innovative responses to the changing customer and market needs (Helfat et al., 2009).

First, let us try to understand what researchers refer to by organisation's capability. According to the RBV, the capability of the organisation is demonstrated in its ability to accomplish against competition and is dependent on the resources it can utilise (Teece & Pisano, 1994). Helfat and Peteraf (2003) provide a similar but clearer definition, which is the ability of an organisation to perform a coordinated set of tasks utilising organisational resources to achieve a particular result. The above two descriptions seem to be limited to the capabilities that manipulate tangible assets. However, as reported by Teece and Pisano (1994) researchers extended this view about resources to encompass organisational tangible and intangible assets. Intangible assets include skill acquisition, the management of knowledge, know-how and learning (Teece, Pisano, et al., 1997), while tangible can be physical, human and organizational (Eisenhardt, 2000).

Second, we come to the question of what qualifies a capability to be dynamic. For a capability to be "Dynamic", it must allow management to effectively coordinate, integrate and reconfigure internal and external organisational resource base to match the changing business environment (Teece, Pisano, et al., 1997). The emphasis here is on the capacity of the organisation to both alter the resource base and at the same time achieve congruence with the changing business environment. Hence, the focus here is not only on providing superior products and services, but more on the capabilities that allow the organisation to sense changes in the business environment and provide innovative products and services in response to these changes in a short period. It is worth mentioning here that "innovative" does not imply superior products but different products that needed to survive given market changes. This view is explained by (Helfat et al., 2009). Literature also noted that Dynamic capabilities could benefit businesses operating in environments that do not experience rapid environment changes (Helfat et al., 2009). This view stems from the fact that dynamic capabilities encourages organisational learning through which the firm can systematically modify and improve its operating processes, hence achieving organisational effectiveness (Zollo & Winter, 1999). However, not all dynamic capabilities improve operating routines, it is particularly those capabilities that enable the firm to sense strategic opportunities and the nature of changes in the market (Helfat et al., 2009).

2.5.4 DCs and Organisational Processes

According to Teece & Pisano (1994) the strategic dimension of a firm is its managerial and organizational processes that encompass dynamic capabilities.

> *The competitive advantage of firms stems from dynamic capabilities rooted in high performance routines operating inside the firm, embedded in the firm's processes, and conditioned by its history."(Teece & Pisano, 1994).*

The conceptualisation of dynamic capabilities as organisational processes rather than generic competencies is evident in literature. Teece, Pisano, & Shuen (1997) define dynamic capabilities as organisational structures, managerial processes and patterns of current practice and learning to support the enterprise productive activity. Similarly, (Eisenhardt, 2000) refers to them as organisational routines and processes embedded in firms, that can integrate, reconfigure, gain and release resources and are driven by best practice. Zollo and Winter (1999) describe them as specialised organisational processes used to improve the performance of the organisations through upgrading and improving operational processes. Helfat et al. (2009, p. 2) confirm this view and report examples of dynamic capabilities as product development, alliances and acquisition, resource allocation, knowledge transfer sand replication routines

Moreover, researchers have emphasised that DCs require continuous refinement and upgrading because processes can lose their value if they support a competence that has lost its value in the marketplace (Teece & Pisano, 1994; Zollo & Winter, 1999). In his earlier publications, he described them as a set of learned processes and activities that enable a company to produce a particular outcome and entrepreneurial activities inside the firm that need to be developed gradually. Hence, dynamic capabilities are strategic activities demonstrated by enterprises that continuously upgrade their operating processes through learned patterns and systematic practices dedicated to process improvement (Zollo & Winter, 1999).

Researchers argued that dynamic capabilities and therefore competitive advantage rest on effective organisational processes but is conditional on the existence of two important requirements. Firstly, organisational processes must be persistent processes in the organisation and not disjoint practices adapted for a short while to deal with a specific business situation. Through such practices organisation can always produce innovative products via a stable product development process (Zollo & Winter, 1999). Secondly, dynamic capabilities can only generate profit and provide competitive advantage if they are based on competences that are difficult to imitate by competitors (Teece, Pisano, et al., 1997). Teece, Gary, and Shuen (1997) further argue that replication by competitors of all the processes that support a certain competence are difficult because processes are complex, dependent on the organisation culture and on a set of other processes.

2.5.5 Difference between Dynamic and Operational Capabilities

Literature classified organisational capabilities or processes into operational and dynamic (Helfat & Peteraf, 2003; Teece & Pisano, 1994). It is important to distinguish the differences between these capabilities in order to avoid confusion about whether a given capability should be operational or dynamic.

(Zollo & Winter, 1999) define operational capabilities as processes with known procedures such as the processing of customer order. This definition agrees with (Helfat & Peteraf, 2003) who described them as a collection of processes to execute and coordinate the development of particular task such as manufacturing of a product. Therefore, operational capabilities are processes that enable the firm to perform the ongoing day-to-day business activities(Helfat et al., 2009). Such processes can be the source of best practice that typically start in one or two firms and then spread to the entire industry (Teece, Pisano, et al., 1997).

Dynamic capabilities are processes that provide the organisation with the abilities to find new forms of competitive strategies to respond to the changing business environment (Teece, Pisano, et al., 1997). Literature has classified DCs with respect to their role. Firstly, those learning capabilities that enable the firm to sense rapid and unpredictable market changes. Secondly, those that can modify the organisation resource base, a capacity that does not apply to operational capabilities (Helfat et al., 2009). These sensing processes could instigate modifications on operational routines to enable adaption to external environmental changes (Zollo & Winter, 1999).

These classifications of dynamic capabilities made researchers refer to them as (high-level) routines because they are able to change operational capabilities and change their own (Dangol, 2012). In contrast, literature described operational capabilities as static (zero-level) because they lack the ability to change their own or other capabilities. Dangol (2012) challenged this view referring to some empirical studies, which showed that even operational capabilities are capable of changing their own and are able to cause changes in other capabilities.

Researchers recognise that both operational and dynamic processes are important for a firm's survival and both have different effects on the organisation. Zollo and Winter (1999) stated that effective operational routines are a necessity, and superior operational routines are a source of advantage. Market dynamism is what dictates whether an organisation should focus on one from the other. According to Zollo & Winter (1999) in a relatively static environment, adequate operational processes can provide the organisation with extended advantage and the organisation can survive without the need for sensing capabilities. However, in situations accompanied by rapid market changes, static operational routines become hazardous and the organisation is then in need for sensing processes to survive. When the change is unpredictable, there is a critical need by organisations to modify their operational as well as dynamic processes. Eisenhardt (2000) extended this view in her discussion about the influence of market dynamism on DCs. She concluded that in a moderately dynamic market where change occurs quickly, effective DCs are dependent on existing knowledge. While, in highly dynamic markets where change is non-linear and unpredictable, DCs are dependent on new knowledge created for the situation. This view reflects the important role of Sensing processes when market dynamism is high.

2.5.6 Sense and Respond Model

The existence of a global market highly dominated by international commerce created a business environment wide open to opportunities and threats (Teece, 2007). This poses critical challenges on small software firms as they operate with limited resources and are often highly dependent on niche markets and big players and are particularly vulnerable to discontinuous or changing business environment (Mathiassen & Vainio, 2007). This environment demands a new business model that attempts to "sense" new market changes and opportunities as they happen, and respond to them quickly. This detection of new opportunities and threats is vital to the survival of enterprises, allows them to sustain or extend its competitive advantage (Teece, 2007).

Teece (2007) identified a number of factors that contribute to the success of the enterprise among which is the adoption of best practice, the delivery of innovative products and services, through (1) sensing opportunities and threats, (2) seizing opportunities and (3) transforming them into valuable outcomes through enhancing, combining, protecting and reconfiguring tangible and intangible assets. Other researchers referred to them as learning processes and described them as instruments for improving operational routines as well as DCs in response to the rapid and unpredictable market changes (Eisenhardt, 2000; Helfat et al., 2009; Zollo & Winter, 1999). They are central mechanism for adding, shedding and renewing resources and operational capabilities (Eisenhardt, 2000; Teece, Pisano, et al., 1997).

The importance of applying the "sense and respond" framework is evident in the work of Mathiassen and Vainio (2007). They have applied Haeckel's sense-and-respond approach on two firms with different maturity levels to evaluate the effectiveness of the dynamic capabilities used by each firm to respond to changes in their business environment. The outcome of Mathiassen & Vainio's study revealed very helpful insights that help in enhancing the model proposed by this study. Mathiassen & Vainio found the maturity level of the firm's management practices and processes are enablers to the success of its dynamic capabilities.

Adopting the sense and respond approach can also help SSEs develop external relationships to compensate for their limited resources. Literature reports that software organisations tend to limit the adoption of sense and respond practices to the project level to help managers respond to customer needs, thereby overlooking the need to bring them up to an organisation level (T. Schmidt & Mathiassen, 2009). The Firm adaptability can be increased by combining project and frim-level capabilities (T. Schmidt & Mathiassen, 2009).

In addition to the sense and respond practices firms can increase their ability to respond to environment changes by adopting a culture of communication and knowledge sharing. Most approaches of knowledge creation and sharing for small software development firms focus on technological aspects and neglect knowledge sharing, and integration at firm level and with the external environment (Mathiassen & Vainio, 2007).

2.5.7 How Dynamic Capabilities Applies to this Study

Dynamic Capabilities literature introduced the sense and respond model, which demands organisations to continuously sense opportunities and threats, and quickly respond to them. This study adapts a similar model to enhance the proposed framework of practices. DC activities have been included in four out of the five capabilities of the model proposed by this study. Following are the four types of DC activities included in the model:

1) **Sense** to identify opportunities and threats by exploring technological opportunities, scanning and evaluating the markets, listening to customers, along with scanning the other elements of the business environment and

quickly carrying out internal and external transformation ahead of the competition.

2) **Seize** by mobilizing resources to address the opportunities.
3) **Transform**: to maintain competitiveness through continual improvement.
4) **Enabler**: is an importance capability/process to establish dynamic capabilities.

2.5.8 Findings of Dynamic Capabilities Literature Review

The review of Dynamic Capabilities (DCs) literature highlighted the important role DCs can play in empowering SSEs with capabilities that can improve their effectiveness and efficiency and can be a source of competitive advantage. This form of competitive advantage requires the firm to control, own or have access to capabilities that provide rapid and innovative responses to the changes in the external environment including customer and market needs.

Achieving competitive advantage rests in the development of effective operational and dynamic organisational processes that are persistent in the organisation and not just disjoint practices. The review has highlighted the differentiation between operational and dynamic capabilities. The literature suggests that the need for dynamic practices is dependent on the market dynamism. In a high changing environment, dynamic processes are very important to the survival of the firm, whereby in markets with low dynamism there is more need for sound operational processes.

Literature also noted that DCs could benefit businesses operating in stable environments that do not experience rapid environment changes because DCs promote organisational learning, which results in continuous improvement of the enterprise processes.

The DCs approach is essential to ensure enterprise agility and survival as markets and technologies change. It is argued that the inclusion of DCs will empower and enhance the proposed model.

CHAPTER 3 - CAPABILITY MODEL

3.1 Introduction

The objective of this chapter is to develop a framework for identifying a set of capabilities that would help SSEs adopt processes that maximise an organisation's capabilities and align it within its strategic direction. The adoption of these capabilities is influenced by many factors such as market dynamics, the type and complexity of the produced software, customer needs and the culture that exists within the organisation. The developed capability framework is intended to provide a taxonomy for analysing and assessing a number of SPI standards.

In addition, the chapter outlines the diverse paths that software development organisations follow and proposes a software development lifecycle that structures the different activities performed by small software organisations and maps the capabilities to the different phases of the proposed SDLC.

3.2 Organisation's Capabilities

The different SPI standards cover different areas of software processes and seek to increase the maturity of the organisation and maximise technical and organisational capabilities. As has been discussed in Chapter 2, SSEs undertaking software development projects need to possess other different capabilities in addition to the technical capability.

Developing a wide array of capabilities through management led initiatives is a critical driver underlying the success of SSEs. It allows SSEs to deepen their expertise in the development of software products. A number of research articles have adopted the concept of capabilities, but there is no common definition for the term. Rousseva (2007) has recognised the development of technological capabilities as crucial to enable software organisations in countries such as China, India and Brazil to enter international markets. Tschang (2001) reported a number of organisational and business environment factors that can help firms to build their capabilities. These factors include existing infrastructure, corporate strategy, an environment supportive of entrepreneurship, human resource base and labour supply constraints.

In this study, the concept of capability is defined as a set of processes that specify requirements of work practices that must be adhered to in order to be compliant with accepted practice. The gradual development of capabilities will transfer the organisation from a state of immaturity (undefined processes) to maturity (standardised and institutionalised processes).

The discussion in the following sections will firstly identify the different paths that a software development organisation can take, and then propose a generic system development lifecycle (SDLC) that is likely to be undertaken by SSEs. The SDLC will be used later to select international standards that are needed for the development of the capability model. Moreover, a discussion will be provided to explain how the proposed capabilities were derived. Finally, an overview of the standards that are within the scope of this study is provided along with an example of how the comparison of processes from these standards was performed to suggest processes for each capability.

3.3 Types of Software Development Activities

Rousseva (2007) defined a number of paths that software development organisations undertake to conduct their business. These include: i) developing own software products and services for domestic market ii) developing software products and services for international market; iii) performing part of the software activities such as requirement analysis and design and outsourcing other activities such as coding, testing and maintenance to international organisations. In addition, software development firms either provide complete solution or a customised solution meeting specific customers' needs (Davis & Sun, 2006). Niessink & Van Vliet (2000) reported that organisations can be contracted to maintain existing software or a component of the product to correct faults, improve product attributes or adapt it to a changed environment. They have identified four types of software maintenance: 1) corrective maintenance to repair bugs and faults; 2) adaptive maintenance for adapting a software to changes in the organisation's environment; 3) perfective maintenance which deals with enhancing the software functional requirements and 4) preventive maintenance aimed at increasing the software maintainability such as updating documentation or adding comments. A survey conducted on SSEs in Queensland, Australia by A Cater-Steel et al. (2006) reported that SSEs supply in house developed bespoke software applications and systems, customise commercial off-the-shelf (COTS) packaged software as well as provide system design, implementation and development of IT services.

The path that a software organisation follows and the type of product developed determines the areas that require particular attention. For example, if the organisation outsources for the international market and produces customised solutions, then there is a need to focus on acquisition and supplier management processes and requirement management, as they will need to incorporate into the supplier contract the product requirements. Those that are carrying out the production of the software will need to focus on the development and release phases of the product's lifecycle. On the other hand, organisations providing after sale software maintenance services must run their business as service providers and focus on processes that relate to the quality of software maintenance (Niessink & Van Vliet, 2000).

There may be other reasons for SSEs to implement particular processes. Drawing on research reports, the most apparent reasons are compliance with regulatory requirements and the perceived value that the process can bring to the organisation, as some processes are seen to have higher value than others, also the ease of implementation and cost (Wilkie, McFall, & McCaffery, 2005)

This study proposes a capability model that benefits SSEs developing software for domestic and international markets. It targets organisations that supply software applications and are involved in software activities that encompass the entire lifecycle model starting from the collection of requirements through to release and maintenance.

3.4 Software Development Lifecycle

It is useful to provide a generic project lifecycle for categorising the different types of activities performed by software organisations and dividing them into phases, each requiring a different set of knowledge and skills and the contribution of different SPI standards. This study proposes the following lifecycle model as a model suitable for SSEs. The lifecycle model will help us later in the identification of the capabilities needed for SSEs, as shown in Figure 3-1 on page 55.

1. **Market Analysis Phase**: The market research phase is influenced by the strategic direction and planning of the organisation. This phase determines the market needs and demand, develops networks with customers and suppliers and predicts the financial needs for developing software products. During this phase, SSEs identify and approach potential customers or suppliers for outsourcing products. At the end of this phase a decision can be made to develop a product or outsource it to an external supplier.

2. **Project Feasibility Analysis Phase**: During this phase, an organisation generates and evaluates alternative solutions, assesses the client's needs and determines whether the solution under analysis is financially and technologically achievable. Operational considerations such as resources, constraints, schedule, culture and legal aspects are validated.

3. **Contracting Phase**: There are two types of software projects that SSEs tend to pursue. Firstly, developing or modifying software or systems to satisfy customer needs or meet contractual requirements, and secondly, contracting a supplier to obtain a system that meets the agreed requirements. For both scenarios, an agreement process must be established planned, executed, monitored and closed. At the end of this phase the organisation signs the agreement with the client or supplier and initiates the project.

4. **Planning Phase**: The objective of this phase is to devise workable plans to meet the desired business requirements. This phase involves in addition to developing the

project plans, getting commitment to the plans from sponsors, extensive interaction with stakeholders, and monitoring that all the efforts that are carried out, are in accordance with the plans. An integrated change control process is required to evaluate and manage the changes through the project and the impact on other projects.

5. **Product Conceptualisation and Development Phase**: This phase involves the understanding of customer requirements and validating that they have been met. It also includes the coding, testing and user acceptance of the product. Iterations of requirements analysis, developments and testing occur until the product is developed and ready for release.

6. **Product Release Phase**: This phase involves the release of the product to the customer for use. Product release is performed in multiple stages depending on the agreement between the customer and software supplier.

7. **Product Maintenance and Servicing Phase**: This is an after release phase and is considered the most expensive and longest phase (Ahmed, 2006). This phase involves a customer support function, which is the single point of contact that receives customer complaints, resolves after sale problems and fixing of bugs, responds to simple GUI and feature enhancement requests, and performs analysis of product performance. It has been reported by research studies that organisations are considering outsourcing their maintenance and support activities to gain competitive advantage (Ahmed, 2006).

8. **Product Retirement**: The goal of this phase is to remove the product from the production process or stop the after sale support of the product. A decision can be taken to stop the production of a product due to varying operational reasons. This must be done with minimal impact on the organisation. In such a case, the system will go through an evaluation process to assess whether the product requires after sale support or whether it can be retired.

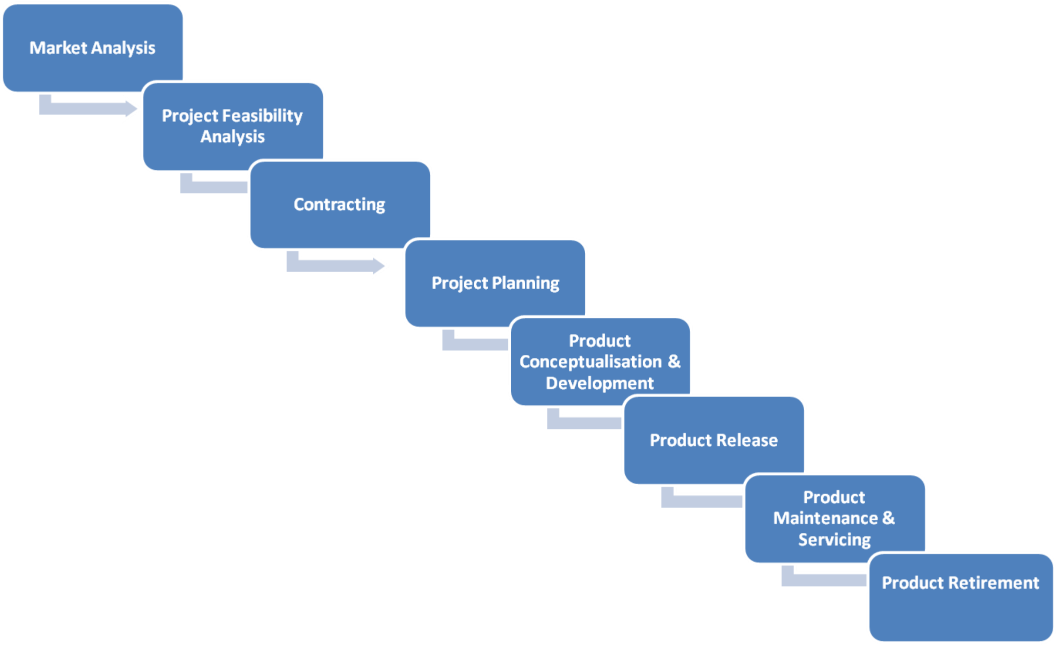


Figure 3-1: Proposed software development life cycle

3.5 Proposed Capability Model

To ensure a coherent organisational environment as suggested by Cattaneo et al (2001), SSEs must take a holistic approach to make sure SPI implementation is successful. To attain a holistic approach this study proposes a capability model for guiding the establishment of processes and for perusing SPI initiatives. The model consists of five capabilities regarded as important for the success of SSEs and critical for their survival as depicted by Figure 3-2. The justifications for the capabilities are discussed in the following sections.

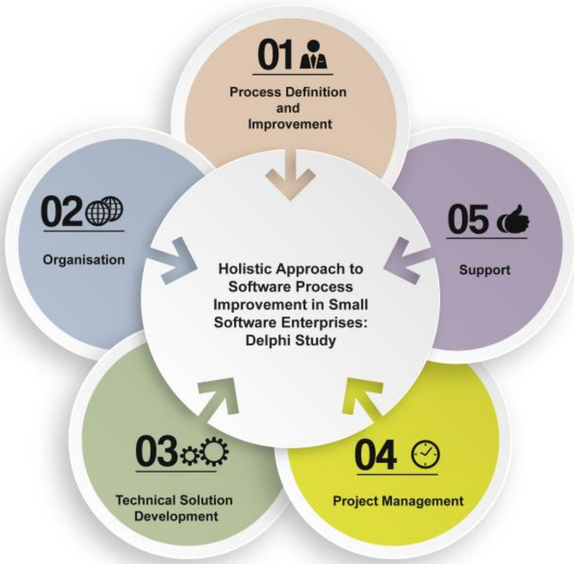


Figure 3-2: The proposed five capabilities

Rousseva (2007) divided the capabilities of a software development organisation into three: a) basic, which includes software programming, testing and quality assurance; b) intermediate, which includes project management, software process management and various operating environment and network applications. Finally, the advanced capabilities, include software, design, developing specialised expertise in a particular domain and research and development.

The contribution of this study differs from the identified capabilities by earlier research. Although previous studies proposed a comprehensive set of capabilities, the identified capabilities were not specific to SSEs. This study identifies capabilities that will help SSEs to obtain the capacity to develop sound processes that underpin the quality of the software product and software projects and at the same time promote successful SPI initiatives.

3.5.1 Capability 1- Process Definition and Improvement Capability

Research studies have reported that SSEs are characterised as having informal processes and administrative behaviour as there is an assumption by the IT community that establishing and defining processes in SSEs is time consuming and costly (Wangenheim et al., 2010). However, empirical studies report that it is possible to define possesses in SSEs in a cost effective way, considering the firm's business objectives and available resources (Wangenheim et al., 2010).

The benefits of having standardised processes in software development organisations have long been acknowledged by the literature (Wangenheim et al., 2010). Yu-Chih Liu et al. (2008) report that it can improve the project outcomes by imposing better controls through more structured activities and by applying learned procedures for developing the software product.

Additionally, SPI standards have been largely based on Edwards Deming's cycle, plan-do-check-act, which states that processes should be defined, implemented and measured to identify the causes of the software product deviations from the prescribed requirements. Therefore, Deming recommended that business processes undergo continuous improvement to achieve the required process goals.

Longitudinal research studies reported that it was impossible to assess the results of improvement actions as most assessed firms showed a low process maturity level and lacked reliable historical and detailed quantitative data (Dalcher, 2009). The absence of quantitative data resulted in empirical studies being totally dependent on objective data collected from the interviews of managers and software engineers or adopting a questionnaire survey method (A Cater-Steel et al., 2006; Dalcher, 2009).

In light of the overview presented above and the findings of the literature review in chapter two, ***process definition and improvement*** was identified as a critical capability in the model.

The objective of this capability is three fold:

a) To plan, implement, and deploy a set of organisational processes that improve organisational and project outcomes.

b) To establish these processes through the development of policies and procedures and employee training programmes.

c) To develop and sustain measurements that will enable the organisation to gain insight into their performance and assess the effectiveness of their processes. The scope of collected measurements will be at project, product and process level. They will be collected and then analysed and compared against identified thresholds to determine whether quality improvements have occurred. These measurements are used as guidelines for continuous improvement of various software processes.

The processes underpinning this capability form the organisation's process assets. Moreover, this capability will be responsible for the adoption of newly deployed processes and for maturing existing processes that need to be improved.

3.5.2 Capability 2 - Organisational Capability

The review of the literature in chapter two revealed that SSEs are focused on operational effectiveness rather than strategic management (Pryor et al., 2010). It has also highlighted that strategic planning can promote organisational entrepreneurship, where entrepreneurial orientation and the experience of employees were identified in the literature as factors that contribute to the success of the organisation particularly SPI initiatives (Wood & Michalisin, 2010).

Moreover, empirical studies stressed that the success of SPI initiatives is attained by achieving internal coherence between the technical, social and organisational aspects of a company (Dalcher, 2009). By demonstrating internal coherence, SSEs will be able to run appropriate and adequate improvement plans (Dalcher, 2009). External coherence is underscored by the literature of dynamic capabilities through the need for SSEs to be particularly responsive to customers' evolving needs and the changing external environment to achieve competitive advantage (Priem & Butler, 2001).

Therefore, the above findings of the literature review discussed in chapter two, demonstrated the need for SSEs to develop the ***organisational capability.*** The objective of this capability is to introduce processes to support the strategic vision and direction of the business and to achieve alignment with the technical product development processes in the organisation leading to the development of successful products and services.

The relationship between organisational and technical capabilities is recognised by the literature (Cattaneo et al., 2001; Rousseva, 2007). According to Rousseva (2007), organisational capabilities underpin the development of technical capabilities. In his study, Rousseva has identified six basic organisational capabilities focused on communication, decision-making, coordination and control, which he believes software organisations are very likely to lack.

Among reported organisational practices are strategy and business development, which contribute to the organisation's strategic planning, and business development (Davis & Sun, 2006). Strategic planning defines the strategic vision and direction of the organization and the managerial decisions to pursue them. Business development comprises entrepreneurial practices that link the organisation with its external environment. This involves developing and maintaining partnerships and networks, understanding the customers and marketplace and expanding into new markets and building knowledge of technologies (Davis & Sun, 2006). It has been recognised that developing business development capabilities is critical to the success and growth of SMEs (Davis & Sun, 2006).

Moreover, the firm's organisational capability must also focus on the human resource aspect of the organisation through careful selection of staff and proper professional development of skills and competencies.

Finally, there is a need for SSEs to establish an organisational culture that facilitates learning. Studies have emphasised that organisations should develop a learning environment by establishing a flexible structure that allows active exchange of information and ideas (Rousseva, 2007).

3.5.3 Capability 3 - Technical Solution Development Capability

These are the technical engineering capabilities utilized in the management and production of the software product or service. This area is expected to be mature as it is well supported by SWEBOK and the majority of SPI standards. Traditional software development environments require managing the software development lifecycle (SDLC) from requirements through to the deployment of the product.

Software engineering literature has long identified the need for technical solution development processes and has paid special attention to the areas of software requirements and testing (Rousseva, 2007). Therefore, the decision to include the ***technical solution development capability*** in the model was not a hard one.

3.5.4 Capability 4 - Project Management Capability

Research emphasises the need for effective *project management* practices for the success of software development projects; projects that do not follow a defined project management methodology or process are likely to fail (McHugh & Hogan, 2011). Research studies also reported that project management practices are adopted much more in SSEs than is assumed by the IT community (Turner, Ledwith, & Kelly, 2010). The adoption of project management processes by SSEs is reported by the literature (Wilkie et al., 2005). It is also reported that organisations are moving towards internationally recognised methodologies rather than internally developed approaches (McHugh & Hogan, 2011).

Among the characteristics that have been reported by the literature on SSEs concerning project management capability is that a number of SSEs have developed this capability, while others have not, and further, the adopted practices were not practised in a consistent manner within the same organisation across different projects (McHugh & Hogan, 2011).

Finally, it is generally accepted that adopting effective project management practices can promote detailed planning. Planning is recognised by Lussier and Corman (1996) as a factor significant contributing to the success of an organisation or in this context projects; Lussier's model is presented in Table 2-3 on page 23.

In light of the above discussion, ***project management*** was introduced as one of the five proposed capabilities that form the structure of the proposed model. The objective of this capability is to deliver the product on time within the agreed budget and scope. Additionally, this allows the mobilisation of the necessary resources (human and others) including the recruitment of needed expertise. It also ensures that the contractual agreement with the product acquirer is performed in accordance with the agreed terms. On the other hand, in the cases where the software development is subcontracted, this capability ensures that a mechanism is in place to select qualified subcontractors and establish, manage and monitor the acquisition of the software product.

3.5.5 Capability 5 - Support Capability

This study argues the need for the ***Support*** capability. This capability is mainly concerned with the transfer of software from development to operations, which is traditionally referred to as the maintenance phase of the software development lifecycle. As reported by Rashid, Wang, and Dorner (2009), there is empirical evidence that demonstrates the need for SSEs to develop this capability separately from the software development capability in order to support and maintain their products after sale. Researchers have considered software maintenance as having more service-like aspects than development as the value of this phase is not the product itself but, similar to a service, it is the activities performed that will result in benefits for the customers (Niessink & Van Vliet, 2000).

This study agrees with the proposition of Niessink and Van Vliet (2000) who suggested the need for software organisations to treat the support activities, after the product is being released, as a service provided to the customer. This proposition is not widely recognised by either the industry or researchers (Niessink & Van Vliet, 2000). The literature stresses the importance of focusing on the managerial side of this capability as well as the technical side (Niessink & Van Vliet, 2000; Rashid et al., 2009) .

Based on the review of software maintenance literature, this study adopted the ***Support*** capability. This capability will include processes to manage service level agreements (SLA) as they are important to guide the communication between the customer and service provider (Niessink & Van Vliet, 2000) . Moreover, it will facilitate the application of an appropriate help desk function supported by efficient incident and problem management processes to manage customer requests for changes and to analyse and resolve problems. This capability needs to also include lightweight configuration and change management processes suitable for SSEs to provide operational visibility and support the decision making process. Change management can be applied to activities at organisational, project and product levels (Lam & Shankararaman, 1998; Rousseva, 2007). The role of the help desk to support this capability by providing the necessary collected data to analyse and resolve problems and incidents is demonstrated by case study research (Niessink & Van Vliet, 2000).

3.5.6 Mapping the Capability Model to the SDLC Phases

Figure 3-3 demonstrates how the five capabilities can be utilised in the different phases of the software development lifecycle discussed in section 3.4 above. The diagram will help us conceptualise the relevance and role of each of the proposed capabilities to the lifecycle of a software product from inception to retirement. Moreover, this diagram will also help in the selection of international standards that are within the scope of this study, which will be used to select processes for each capability.

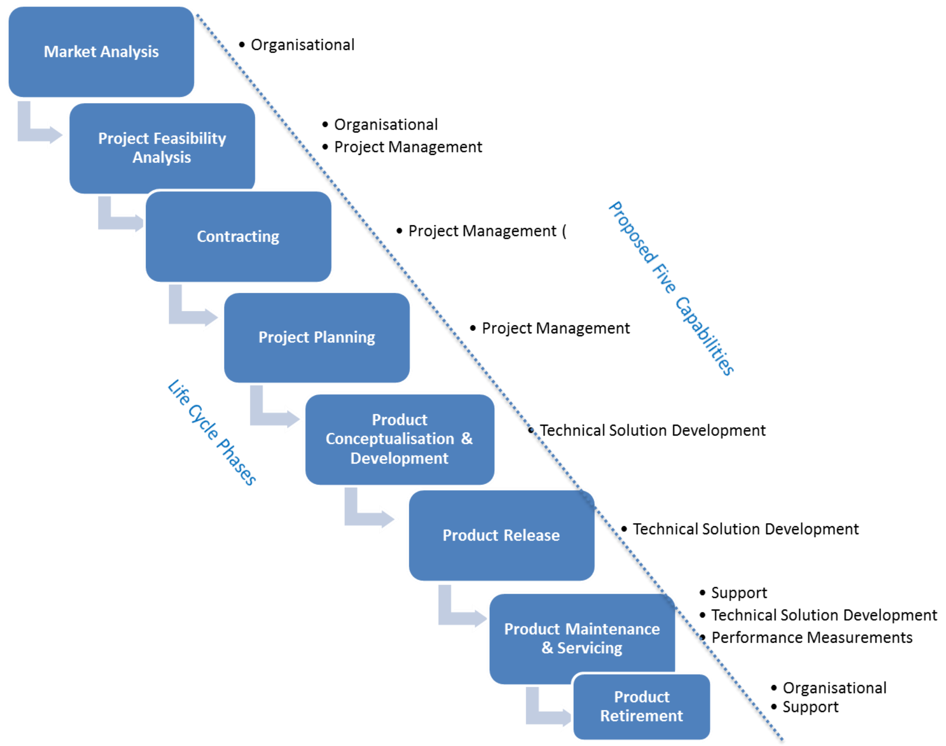


Figure 3-3: software development phases mapped to the proposed capability model

3.6 Scope of the Study

This study targets small software development organisations that are involved in the supply of software products. A typical software organisation will go through all or most of the lifecycle phases discussed in section 3.4. Therefore, the scope is limited to software organisations that are involved in the development of the software and the aspects of services needed to maintain and support the product after release. The services that are within the scope of this study are typically the activities that belong to the maintenance phase of the proposed lifecycle model, as opposed to the services provided by IT service organisations, where services are considered their main product delivered to the customer.

Software process improvement involves various types of standards that guide the organisation's SPI endeavour. This study divides the standards into different groups depending on the role the standard plays in improving the project and product outcomes. Each standard has its own focus areas and processes with overlapping areas and processes with other models. Following is the categorisation of the standards that are relevant to building the proposed capability model.

- Project Management Models: identify a set of knowledge areas, techniques and tools to manage the activities of a project to create a unique product or service.
- Process Reference Models: provide a definition and description of the processes that represent the key areas of practices that have impact on SPI outcomes.
- Assessment Models: guide the performance of assessments by providing a framework for assessing the organisation by maturity level or process capability level.
- IT Service Management Models: frameworks of best practice that attempt to deliver and support IT services to meet the changing business needs.
- IT Governance models: identify a set of processes that ensure the delivery of the IT outcomes in a controlled manner that helps enhance organisation success and mitigate risk.

Figure 3-4 provides a clearer presentation of the standards and category areas. Although the diagram presents a set of standards including COBIT; however COBIT is excluded from the scope of this study, and the set of standards listed below were identified as relevant to this study:

1. Software Engineering Body of Knowledge (SWEBOK).
2. A Guide to the Project Management Body of Knowledge (PMBOK Guide)
3. CMMI for Development, Version 1.3 (CMMI-DEV V 1.3)
4. CMMI for Acquisition, Version 1.3 (CMMI-ACQ V 1.3)
5. Software Process Improvement and Capability Determination (ISO/IEC 15504) also known as SPICE
6. Software life cycle processes (ISO/IEC 12207, 2008)
7. Information Technology Infrastructure Library (ITIL) Version 3,

Figure 3-5 presents how the IT standards (in-scope) can be best utilised at the different phases of the software development lifecycle. This suggestion takes into consideration the processes covered by each standard.

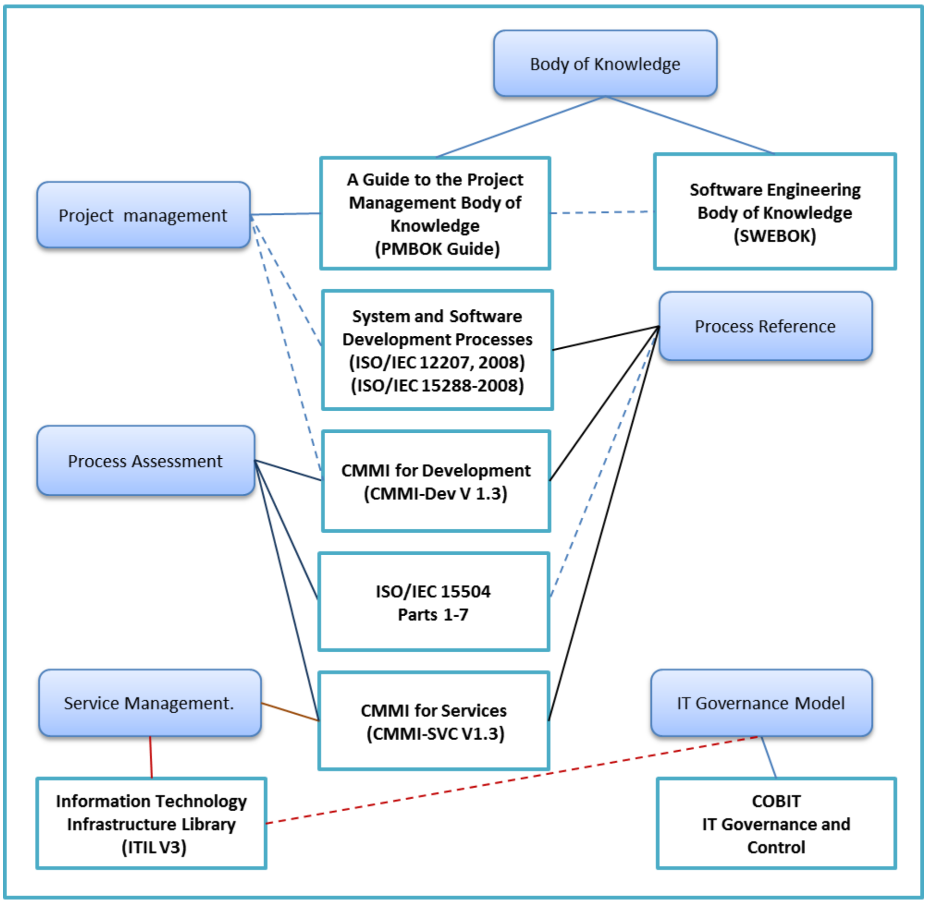


Figure 3-4: Categorization of IT standards

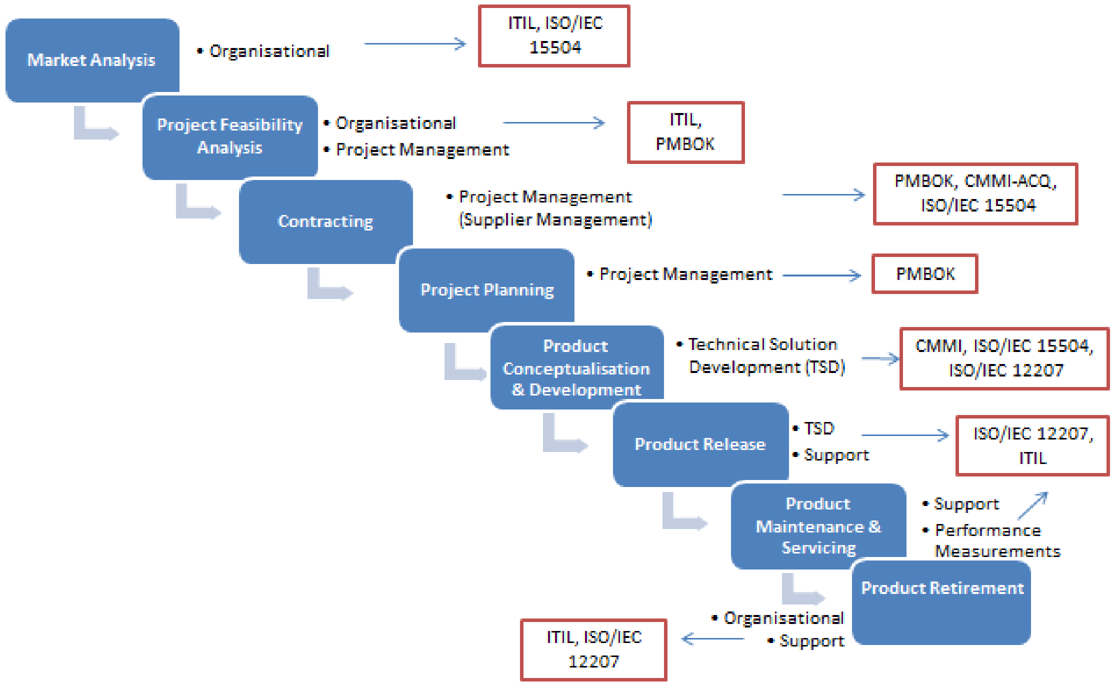


Figure 3-5: The role of SPI standards on the SDLC

3.7 Comparison of Standards

The main motivation behind this section is to provide insights into how the standards have been utilised in the development of the model.

The five capabilities were used as a taxonomy for performing a comparative analysis between the standards identified in section 3.6. For each capability, the relevant standards were selected and a comparison was performed to identify the processes that underpin the capability. This comparative analysis if published, can give SSEs insights into the practices that will benefit them, the areas where the standards overlap and how the standards could complement each other. However, the detailed information related to the comparison has been removed from this study document due to word limitation requirements and only one short example is presented in the following section.

It is also worth to justify here why the software engineering body of knowledge (SWEBOK) was included in the comparison. SWEBOK is the body of knowledge for software engineering, which means that it is a document that focuses on the knowledge, skills and practices that must be applied by software organisations and practised by software development engineers.

It is not expected that SWEBOK can be a comprehensive document of software engineering knowledge areas, as is stated explicitly by the authors of the guide, its objective is to describe the core body of knowledge. Yet, it still serves as a guide for the established practices recommended by industry. Additionally, the inclusion of SWEBOK will help in assessing how far each capability is covered, as we believe they are core competence areas for SSEs to improve the quality of the product. SWEBOK also identifies the accepted software engineering knowledge and skills that are expected to be mastered by practising engineers.

The analysis of the knowledge areas of SWEBOK were not conducted based on the breakdown of topics presented in the Guide. For example, the breakdown of some topics presented by the guide includes fundamental concepts and practical considerations. The basis of the comparison was limited to the processes described in each topic.

This comparison was carried out for all the capabilities to develop the initial version of the proposed model. The process areas related to the selected capability were identified from SWEBOK first, then the relevant standards. The underlying processes from all the standards were compared. The comparison was performed by analysing and comparing the process purpose and outcomes from each standard. Links between processes that have similar purpose were drawn at the process or sub process level. The processes that were included in the majority of compared standards were identified as common practices. These were added to the proposed model after considering their relevancy to SSEs by reviewing relevant research studies in the specified area. Table 3-2. provides an example of how the comparison was performed for process *3.1 Requirements Elicitation*. The comparison for this process included the following standards:

- Software Engineering Body of Knowledge (SWEBOK).

- CMMI for Development, Version 1.3 (CMMI-DEV V 1.3)
- CMMI for Acquisition, Version 1.3 (CMMI-ACQ V 1.3)
- ISO/IEC 15504 Information technology process assessment
- ISO/IEC 12207 Software life cycle processes.

The resulting proposed processes for *3.1 Requirements Elicitation* is presented in Table 3-1. 3.1 *Requirements Elicitation* is the first process of the *3.Technical Solution Development* capability.

Table 3-1: Identified processes for *3.1 Requirements Elicitation process*

3.1. SOFTWARE REQUIREMENTS	
3.1.1 Obtain software Scope	• This is an input from Scope Management (project charter), which identifies the high-level software objectives as approved by the customer.
3.1.2 Elicit requirements	• Identify fact-finding sources and techniques to elicit requirements. • Develop requirement definition document that aims at the business audience, customers and users.
3.1.3 analyse requirements	• Analyse the needs of relevant stakeholders. • Identify detailed functional and non-functional requirements, design constraint and process constraint. • Develop requirement specification document, develop requirements in technical terms necessary for architectural design.
3.1.4 Trace requirements	• Requirements must be compared for traceability and consistency with users' requirements (traceable and consistent). • A requirement must be traceable to some source such as to an authoritative source, whether a person or document. Each requirement should have a unique identifier allowing the software design, code, and test procedures to be precisely traced back to the requirement.
3.1.5 Validate and prioritise requirements	• Frequently review requirements documents with stakeholders. This validation ensures that requirements accurately reflect users' needs (correct, unambiguous and relevant). • Stakeholders must assign a priority to each requirement (e.g. mandatory, critical, desirable or optional).
3.1.6 develop requirements baseline	• Users agree that requirements are correct and complete. The result is a requirements baseline approved and accepted by the customer. This baseline will be an input to the requirement management process (complete and correct).
3.1.7 Develop test cases	• Develop test cases to verify compliance with requirements (testable).

Appendix B provides a table presenting the entire process mapping for all the capabilities.

Table 3-2 Comparison of standards for *3.1 Software Requirements*

SWEBOK	CMMI DEV & ACQ	ISO/IEC 15504	ISO/IEC 12207
1. Software Requirements	**Requirements Management (Project Management Process)**	**5.4 Engineering Process Group**	**6. Technical Processes**
Requirement Elicitation	SG 1 Manage Requirements	**5.4.1 ENG.1 Requirement Elicitation**	**6.4.1 Stakeholder Requirements Definition Process**
Requirement Sources	SP 1.1 Understand Requirements	BP1. Obtain customer requirements and request	6.4.1.3.1 Stakeholder identification
Elicitation Techniques	SP 1.2 Obtain Commitment to Requirements	BP2. Understand customer expectations	6.4.1.3.2 Requirements identification
Requirement Analysis	SP 1.3 Manage Requirements Changes	BP3. Agree on requirements	6.4.1.3.3 Requirements evaluation
Requirement Classification	SP 1.4 Maintain Bidirectional Traceability of Req.	BP4. Establish customer requirement baseline	6.4.1.3.4 Requirements agreement.
Conceptual Modelling		BP5. Manage customer requirements changes	6.4.1.3.5 Requirement recording
Architectural Design and Requirement	**Requirements Development**	BP6. Establish customer query mechanism	
Allocation	**SG 1 Develop Customer Requirements**	**5.4.2 ENG.2 System Requirements Analysis**	**6.4.2 System Requirements Analysis Process**
Requirement Negotiation	SP 1.1 Elicit Needs	BP1. Establish system requirements	6.4.2.3.1 Requirements specification
Requirement Specification	SP 1.2 Transform stakeholder needs into Customer Requi.	BP2. Optimise project solution	6.4.2.3.2 Requirements evaluation
The System Definition Document	**SG 3 Analyse and Validate Requirements**	BP3. Analyse system requirements	
The System Requirement Specification	SP 3.1 Establish Operational Concepts and Scenarios	BP4. Evaluate and update system req.	
The Software Requirement Specification	SP 3.2 Establish a Definition of Required Functionality	BP5. Ensure consistency	
Requirement Validation	and Quality Attributes	BP6 communicate system requirements	
Requirements Reviews	SP 3.3 Analyse Requirements	**5.4.3 ENG.3 System Architectural Design**	**6.4.3 System Architectural Design**
Prototyping	SP 3.4 Analyse Requirements to Achieve Balance	BP1. Describe system architecture	6.4.3.3.1 Establish architecture
Model Validation	SP 3.5 Validate Requirements	BP2. Allocate requirements	6.4.3.3.1 Architectural evaluation
Acceptance Tests		BP3. Define internal and external interfaces	
Practical Consideration		BP4. verify system architecture	
Requirement Tracing		BP5. Evaluate alternative system architecture	**7. Software Specific Processes**
		BP6. Ensure consistency	**7.1.1 Software Implementation Process**
		BP7. Communicate system architecture design.	7.1.1.3.1 Software implementation strategy
2. Software Design	**SG 2 Develop Product Requirements**	**5.4.4 ENG4. Software requirements analysis**	**7.1.2 Software Requirements Analysis Process**
Architecture Design	SP 2.1 Establish Product and Product Component Req.	BP1. Specify software requirements	7.1.2.3.1 Software requirements analysis
Detailed Design	SP 2.2 Allocate Product Component Requirements	BP2. Determine operating environment impact	**7.1.3 Software Architectural Design Process**
	SP 2.3 Identify Interface Requirements	BP3. Develop criteria for software testing	7.1.3.3.1 Software architectural design
		BP4. Ensure consistency	**7.1.4 Software Detailed Design Process**
		BP5. Evaluate and update software requirements	7.1.4.3.1 Software detailed design
		BP6. Communicate software requirements.	
		5.4.5 ENG5. Software Design	
		BP1. Describe software architecture	
		BP2. Define interfaces	
		BP3. Develop detailed design	
		BP4. Analyse the design for testability	
		BP5. Ensure consistency	

3.8 Summary

This chapter identified the paths and activities taken by software development organisations and mapped the capabilities to the software development lifecycle. The major outcome of this chapter is the identification of the five-capability model needed by small software organisations in order to reach a higher maturity level, and deliver successful and consistent software development projects. In identifying the capabilities, the study draws on earlier SPI and software engineering research and the study of international standards.

This chapter further defined the scope of the study and defined the standards that were utilised in the development of the proposed model. Finally, the chapter highlights how the comparison of the standards was performed and provided a short example.

CHAPTER 4 - METHODOLOGY

4.1 Introduction

This chapter discusses the philosophical stance taken and provides the rationale and justifications for the research design choices. It addresses the methods applied and how answers to the research questions have been reached.

Moreover, this chapter provides a discussion of the Delphi method in detail including a description of the process, the selection criteria of panel members and a description and justifications of the quantitative statistical measurements used in analysing the data and reaching a consensus. Finally, the chapter addresses the measures of rigour undertaken to ensure the reliability and validity of the findings.

4.2 Elements of Research Design

Part of the confusion in research design is that, not only are researchers faced with an array of methodologies available to them, but also the terminology and classifications are often inconsistent and contradictory (Crotty, 1998). The following paragraphs demonstrate these inconsistent and overlapping classifications and terminologies used by different researchers.

Creswell (2008) defines research design as a plan or a proposal to conduct research informed by three elements: the assumptions of the philosophical worldview, the strategy of inquiry and the research methods and procedures. On the other hand, Crotty (1998) proposes a design framework that consists of four basic elements that comprise any research process. The framework details and the interrelationship between elements are depicted in Figure 4-2 and Figure 4-1. Other studies have proposed different approaches to research design (Easterbrook, Singer, Storey, & Damian, 2008; Flower, 2009; Higgs, Horsfall, & Grace, 2009).

To demonstrate an example of the inconsistency in terminologies, Easterbrook et al. (2008) listed positivism, constructivism, critical theory and pragmatism under epistemology, while Crotty listed constructivism under epistemology and positivism and critical theory under theoretical perspective.

This study uses Crotty's classification as it provides a clearer structure for applying the different research terms and was found to be more helpful in analysing and understanding the research process. Crotty demonstrates that an interrelationship exists between the framework elements, the methodology that governs the choice of methods, the theoretical perspective that lies behind the methodology, and the epistemology that informs the theoretical perspective.

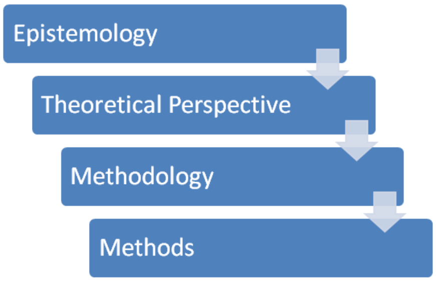


Figure 4-1: Interrelationship between research design elements Cotty (1998)

Figure 4-2 presents Crotty's framework of research design.

Epistemology	Theoretical perspectives	Methodology	Methods
• Objectivism • Constructivism • Subjectivism *(and their variants)*	• Positivism and PostPositivism • Interpretivism ○ Symbolic ○ Interactionism ○ Phenomenology ○ Hermeneutics • Critical inquiry • Feminism • Postmodernism etc.	• Experimental research • Survey research • Ethnography • Phenomenological research • Grounded theory • Heuristic inquiry • Action research • Discourse analysis *etc.*	• Sampling • Questionnaire • Observation • Interview • Focus group • Case study • Visual ethnographic methods • Statistical analysis • Comparative analysis • Interpretive methods • Document analysis • Content analysis *etc*

Figure 4-2: Elements of research design, adopted from Crotty (1998).

Following is a discussion of the philosophical elements adapted from Crotty (1998) to justify the research philosophical stance and methods used to collect evidence.

4.3 Philosophical Assumptions

There are different philosophical assumptions regarding what constitutes scientific truth. In fact, different knowledge can be derived from the same phenomena under study by employing different philosophical assumptions. These assumptions define how we approach the problem, formulate the research questions and the methods we utilise to collect data (J. Creswell, 2012). J. Creswell (2012) stresses that these assumptions are deeply rooted in the discipline we follow and are reinforced by the scholarly community we belong to. To Creswell some disciplines employ diverse research approaches, while others are narrowly focused.

Section 2.2 of Chapter 2 highlights the multi-disciplinary nature of software engineering and stresses the need for multiple approaches of inquiry to address the overlapping areas shared with other disciplines as well as those unique to software engineering domain. This view is shared with A Cater-Steel (2004a, pp. 91-93)

Developing a philosophical stance for research requires making several core assumptions that underpin the chosen research approach. Researchers are encouraged to be explicit about these assumptions in order to determine what they believe and what matches their chosen research methods (Crotty, 1998; Holden & Lynch, 2004).

The first assumption is regarding ontology, which poses questions about researchers' views of reality. Idealism is one form of ontology. An idealist believes that a single abstract truth exists independently from our minds and consciousness, whereas a realist denies the existence of universal truth and sees that reality is a construction of the human mind (Crotty, 1998). The researcher's view of reality is the foundation of all other assumptions.

Secondly, researchers consider their epistemological assumptions, which pose questions about knowledge and how we obtain it (Holden & Lynch, 2004). There is a range of epistemologies. One form is Objectivism, which gives one objective meaning to reality, where that truth is irrespective of our awareness (Crotty, 1998). Another form of epistemology is Constructivism. A constructivists rejects the idea of one objective truth, and see that meanings about reality can be constructed through interaction with other subjects or world realities (Crotty, 1998). Therefore, multiple meanings for the same phenomenon can exist since people construct meanings in different ways. In seeking knowledge, constructivists researchers try to get as close as possible to participants being studied (J. Creswell, 2012). The third form is Subjectivism. Subjectivists, who are at the other end of the continuum, see that truth is very subjective and does not manifest through the interaction with human subjects, but is based on our perception of reality that is influenced by our dreams, culture and beliefs (Crotty, 1998).

Thirdly, the adopted ontological and epistemological concepts underpin the *Theoretical Perspective* or the research approach. The selected theoretical perspective provides context for the research process and brings together a number of assumptions that lie beneath the methodology of choice (Crotty, 1998). These assumptions stipulate the choice of methods employed to answer the research questions. Here a distinction can be made between positivism and interpretivism (Crotty, 1998). As explained by Easterbrook et al. (2008) and Flower (2009), positivism is closely associated with closed experiments, and relies on concrete theories using quantitative methods, while, Interpretivism concentrates less on verifying theories and accepts that knowledge cannot be separated from its human context. Hence, Interpretivists seek qualitative data, collected from focus groups and cases studies.

The above discussion may lead us to think that researchers who adopt an objectivist epistemology must employ quantitative methods, while constructivists and subjectivist must adopt qualitative methods. According to Crotty (1998) and J. Creswell (2012) it is not problematic to employ both qualitative and quantitative methods as long as we remain epistemologically consistent by remaining consistently objectivist or consistently constructivist (or subjectivist) (Crotty, 1998).

As a response to this view, Post-positivism has emerged as a variation to the classical positivism view, which accepts the basic assumptions of positivism regarding ontology, epistemology and the use of scientific experiments. Proponents of this approach accept the employment of a social science theoretical lens through limited interaction with research subjects (J. Creswell, 2012; Crotty, 1998).

4.4 Research Philosophical and Methodological Approach

The phenomenon investigated in this study is a set of essential practices in relation to SSEs and the empirical component of the study attempts to answer the following research questions:

RQ2 What are the specific capabilities critical for the desired model that would benefit SSEs at each phase of the software product lifecycle?

RQ3 What are the underlying processes desired for each identified capability suitable for SSEs?

RQ4 What are potential dynamic capabilities suitable for SSEs?

To answer the above research questions, the study attempts to seek the opinion of a selected group of experts to assess and suggest improvement to the proposed model. Therefore, the Delphi technique was selected as the methodology underpinning this research study. Delphi is considered a rigorous tool for soliciting feedback from experts as recommended by Okoli and Pawlowski (2004).

A Delphi study assumes ontological assumption of single reality on which experts agree. As discussed in the foregoing section a positivist takes the role of an observer not actively involved in the research, focusing on analysing concrete verifiable facts collected about the assessed model. This paradigmatic assumption is supported through the application of the quantitative method of data collection and through the utilisation of statistical measures for the identification of consensus.

However, adopting a positivist position has little to say about the perception of experts regarding the selected practices. Moreover, the Delphi technique establishes a level of interactivity between the researcher and panellists by means of exchanging feedback. This feedback can help the researcher reveal areas of improvement regarding the proposed model and provides participants with rich insights derived from the experience of other participants to help them reform their subjective judgement to reach a consensus.

A key benefit of Delphi technique for this study is the contribution of each participant to the refinement of the proposed capabilities and processes. Hence, it is mainly concerned with opinions, ideas and words, which require interpretation and a certain level of objectivity.

Based on the above assumptions, the research will assume a post-positivist theoretical perspective, which employs multiple realities through critical analysis and interpretations of the varying views to support the statistical findings (J. Creswell, 2012; Crotty, 1998).

This approach is complemented by the adoption of quantitative and qualitative research methods. Multiple data collection methods are justified when one single method is not enough to develop complete understanding about the research problem (Venkatesh, Brown, & Bala, 2013). Collecting qualitative data to support the research findings is as important as objective quantitative data, since it provides powerful insights beyond the presented superficial view of the problem (J. Creswell, 2012; Easterbrook et al., 2008).

The Delphi study undertaken is both quantitative and qualitative in nature through collecting data in multiple iterations. Some consideration is given here to the assumptions underpinning the Delphi method, which are discussed in the following sections. The research philosophical and methodological approach is summarised in Table 4-1 .

Table 4-1: Summary of the research philosophical and methodological approach

Epistemology ➡	Theoretical Perspectives ➡	Methodology ➡	Methods
Objectivism Objective evidence is collected based on concrete facts, gathered from panellists and measured using scientific statistical methods.	**Post-Positivism** Assumes multiple realities, which can be constructed and approximated by employing a social lens.	**Delphi technique** to gain expert consensus on the proposed model. It is a valid approach for this study as the experts bring work experience knowledge, and the feedback between rounds can stimulate new ideas.	**Two-round questionnaires** are sent to a panel of experts. The questionnaire collects quantitative and qualitative data. For data collection, the survey instruments use 5-Likert point scale for the rating of capabilities and processes, as well as open-ended questions for collecting qualitative data. Statistical measures are used for the identification of consensus.

4.5 Methodology

According to Crotty (1998), every piece of research calls for a unique methodology. The choice of methodology underpins the choice of ontology, epistemology and theoretical perspectives. Although a traditional survey could have been used to receive an input on the proposed model, a structured Delphi survey was selected as it is considered a more rigorous tool for soliciting feedback from experts as recommended by Okoli and Pawlowski (2004).

4.5.1 Overview of the Research Process

To elaborate on the justifications for the methodology, it is important to provide a quick overview of the research process shown in Figure 4-3, which consisted of four main steps to investigate the research questions.

Step 1: An extensive literature review to understand the strength and weakness of current SPI literature and issues facing organisations, specifically SSEs in their software process improvement initiatives. A review of statistics on SEs related to various facts, issues and concerns that could have an impact on SPI efforts in those organisations.

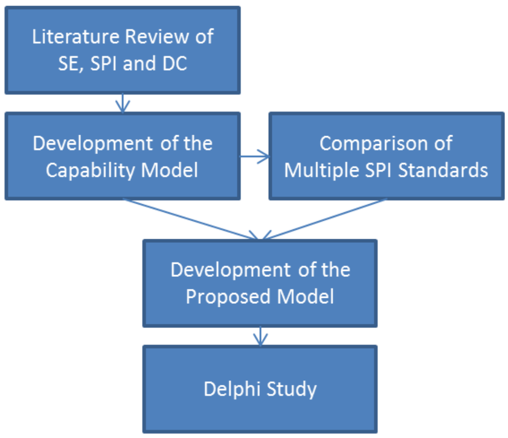


Figure 4-3: Summary of Research Process

Step 2: The development of an array of capabilities that are considered critical drivers for the success of SSEs. The result of this phase is the development of five key capabilities (Process Definition and Improvement, Organisational, Technical Solution Development, Project Management and Support). The identified capabilities formed the structure of the model and served as a taxonomy for comparing the processes proposed by existing standards.

Step 3: The development of an initial process model that the experts could build upon. The initial model is heavily drawn on a comparison of existing software improvement standards (CMMI, IS0/IEC 15504, ISO/IEC 12207, PMBOK and ITIL), refer to section 3.7 for details. Processes from all these models were compared and merged together to form the proposed process model. The outcome of this phase is the first draft of the proposed model.

Step 4: The employment of the Delphi method to evaluate and enhance the processes suggested in the model (2-3 iterations). The output of this step is the approved final model for small software enterprises.

4.5.2 Overview of Delphi Technique

Delphi studies have been employed for a variety of situations including theory generation and for framework or strategy development where the opinions and judgments of experts and practitioners are necessary (Hasson & Keeney, 2011; Okoli & Pawlowski, 2004). They reflect the opinion of a carefully selected sample of experts on the phenomenon under study and the validity of the outcomes are measured in terms of the consensus achieved between those experts. There is a good amount of information systems research that has used the Delphi technique as reported by Okoli and Pawlowski (2004); R. Schmidt, Lyytinen, Kei, and Cule (2001).

Delphi studies differ in terms of application, number of rounds, composition, size and selection of panel members, forms of feedback, techniques applied in reaching consensus, design and the administration of questionnaires. Although the literature presents many types of Delphi techniques, three are mainly in use and these are: classical, policy and decision (Hanafin & Brooks, 2005).

Generally, the Delphi approach is characterised by five fundamental elements (Hanafin & Brooks, 2005; Okoli & Pawlowski, 2004):

1. Adequate panel selection criteria
2. Securing participants anonymity
3. Controlled feedbacks that enable participants to refine their judgement based on the information gathered from the group.
4. Effective survey instrument
5. Suitable statistical measures to identify consensus; the difficulty associated with Delphi studies is relying on reaching an agreement among participants.

In this study, the chosen design is primarily guided by researching the problem in accordance with the ten main Delphi categories identified by Hasson and Keeney (2011). This study employs a modified Delphi, which is characterised by two or more rounds, and where panellists are provided in the first round with a model drawn from different sources. The following sections provide a description of the steps undertaken and an explanation of the rationale for the adopted design choices.

4.5.3 Objectives of Delphi

The Delphi methodology provides an opportunity for evaluating and reaching a consensus on the capabilities and relevant underlying processes that constitute a process model workable for SSEs. Panellist were asked to recommend changes to the model (remove, add processes) to make it feasible for SSEs. Table 4-2 describes how the Delphi method answers the research questions.

Table 4-2: Objectives of Delphi method

	Research Questions	Delphi Study
RQ2	What are the specific capabilities critical for the desired model that would benefit SSEs at each phase of the software product lifecycle?	➢ To seek the opinion of field experts on the proposed capabilities.
RQ3	What are the underlying processes desired for each identified capability suitable for SSEs?	➢ To seek the help of field experts to improve the proposed processes and sub processes.
RQ4	What are potential dynamic capabilities suitable for SSEs?	➢ To seek the help of field experts to improve the proposed model by incorporating DCs into the model.

4.5.4 Rationale for Choosing Delphi Technique

Researchers must always explore all options of research methods for collecting their data. In this study, there was an evident need to seek the opinion of subjects who have a high level of experience in relevant areas in the field, and who are directly involved in software development projects, to assess the adequacy of the proposed practices and evaluate their applicability to SSEs. The utilisation of a technique based on group assessment is recognised as increasing the chances of success for this research study in providing more trusted solution (Hallowell & Gambatese, 2009). The solution would be based on a pool of experienced opinions, which in turn would enhance individual judgement and capture collective opinion on the evaluated model.

A number of group techniques were explored including focus group, nominal group and group interview. All these techniques are workable options to achieve the objective of this study. However, the Delphi technique is found to provide the following advantages over the other techniques.

First, Delphi is a systematic and interactive research technique that helps in collecting rich qualitative data from a group of experienced experts. This fits well with the philosophical approach selected for this study.

Secondly, one of the most important reasons for adopting Delphi is that it is possible to select participants with the relevant experience and knowledge from geographically dispersed locations or international regions. This heterogeneity of participants can assure the validity of results (Hallowell & Gambatese, 2009). Additionally, the model will receive international relevance, besides being limited to local context. On the other hand, the other group techniques require collecting experts in one physical location, which can be costly and is not feasible for the scope of this study.

Thirdly, the anonymity of research participants is an important characteristic of a Delphi study. It is accepted that participants of Delphi studies will express their opinion based on personal knowledge and experience without the influence of other group members (Hallowell & Gambatese, 2009; Von Der Gracht, 2012).

Finally, the Delphi technique can provide an opportunity of running follow-up interviews with subjects from inside or outside the panel, if required (Hasson & Keeney, 2011).

4.5.5 Selection of Panel Members

This section addresses the second step in the Delphi process. A fundamental component of the Delphi technique that differentiates it from other forms of survey research is the use of a panel of field experts. Delphi studies do not require a statistical sample that is representative of the population. The validity of the study relies primarily on recruiting participants who possess the relevant experience and knowledge in the software development field and who could convey a valid opinion on this matter. Therefore, the criteria for the selection of panel members is a critical requirement for the success of the study (Okoli & Pawlowski, 2004).

The selection criteria used to qualify members for the panel in this study is discussed below. This criteria was identified by reviewing Delphi study Participants who did not meet the selection criteria were excluded from the study.

- Participants must have relevant experience and direct involvement in one or more areas of the software development lifecycle, mainly the five capabilities identified in the model. To be accepted as a panel member for this study, participant can be process improvement consultants, project managers, software solution architects, system analysts, or have any other relevant experience. This criterion ensures that a wide scope of opinion is collected and the collective sample of participants can provide input to the five capabilities of the model.
- Some of the selected panel members must have experience in SSEs. This criterion ensures the relevance of opinion to the SSEs context.
- Another important criterion that was considered is participants' willingness to participate. Therefore, only participants who showed high interest and motivation in contributing to the study were selected. This criterion is considered to be a step to a lower dropout rate .
- Participants must also have a minimum of five years of experience and knowledge in software development projects to provide a valid opinion on this matter (R. Schmidt et al., 2001). This criterion ensures the adequacy of members' knowledge and experience and the validity of feedback.

The structure of the panel for this study is discussed in section 4.5.8.

4.5.6 Delphi Process

Theoretically, the Delphi process can run in a number of iterations, for example two or more rounds until consensus is achieved. In this study, the process was structured in two rounds during which open-ended questionnaires were sent to participants for individual comments and for rating of the practices suggested in the initial proposed model. The results were analysed and comments of participants were fed back to panellists for reconsideration. Consensus was mathematically derived and participants comments were qualitatively analysed.

The Delphi process used conforms with the guidelines suggested by Day and Bobeva (2005); Hallowell and Gambatese (2009). Following is a discussion of the specifics of the process and Figure 4-4 provides a summary and visual presentation of the Delphi process.

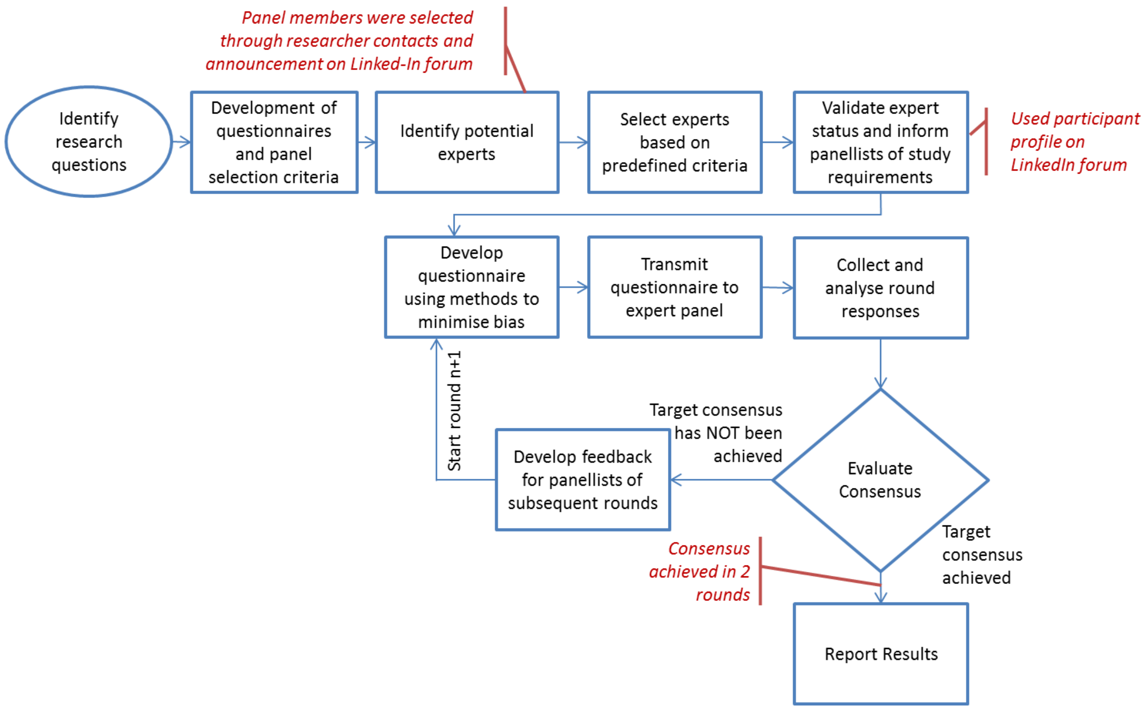


Figure 4-4: Delphi Process derived from Hallowell and Gambatese (2009)

Stage 1 – Preparatory Phase: Some studies refer to this stage as the Exploration Stage which covers the necessary activities before the start of the rounds (Day & Bobeva, 2005). This stage started with the determination of the objectives of the Delphi method. Then there was review of the Delphi literature, followed by the development of the criteria for panel selection, and finally the development of the round-1 questionnaire by transposing the proposed model into a set of questions.

Stage 2 - Identification of Panel Members: The identification of potential participants was through announcements published on the LinkedIn forum. An announcement was posted on relevant LinkedIn group forums inviting members to participate in the study and an invitation letter was forwarded to all participants who requested to be contacted by email (see Appendix C). The following forums were used:

- Software Process Improvement (SP) Consultants
- Software Testing and Quality Assurance
- IT Project, Program, Portfolio Management Experts
- Program Management Academy
- Computer & Software Engineering Professionals
- New Zealand IT industry

Using the LinkedIn forum as a means to reach out to interested participants was very successful as it helped in validating members' backgrounds and experiences through their profile page. The validation of panel members helped in selecting individuals who met the selection criteria of a minimum relevant five years' experience in the field. Studies reported that subjective selection of panel members will lead to a selection bias (Van Zolingen & Klaassen, 2003). A selection bias is recognised as setting up group of panel experts who lack the adequate experience in the field (Van Zolingen & Klaassen, 2003).

Stage 3 – Establishment of Panel: Once acceptance was received, an email was forwarded to every interested subject with a participant information sheet explaining the purpose of the research, expected benefits to the IT community and what is expected from them in terms of time and efforts to complete each round of the Delphi study (see Appendix C). A consent form was also attached to the email, and participants were asked to accept by printing their name and date and then return it to the researcher (see Appendix C). Participants were encouraged to nominate others for inclusion in the panel. Initially fifteen participants agreed to join the panel; however, nine formally accepted by signing and returning the consent form. The remaining six did not meet the deadlines of the first questionnaire and replied to the follow up emails advising that they will not be able to participate due to business commitments. It is reported by the literature that the highest dropout of participants usually happen in the first round as many participants appear to find filling the questionnaire will take longer than they expect (Van Zolingen & Klaassen, 2003).

Stage 4 – Data Gathering (First Round): On receiving the signed consent form, the first questionnaire was forwarded to all nine participants by email. They were given three weeks to fill and return the questionnaire. However, it took six weeks for all participants to respond. Reminders were sent out one week prior to the due date to all participants. Personal reminders were sent to those who had not sent the questionnaire on the due date.

In this round, panellists reviewed the model, provided extensive comments and further suggested improvements. They were asked to rate all the suggested practices for the five capabilities on a 5-point Likert scale with respect to two assessed categories importance and feasibility. Refer to Appendix C for a copy of the questionnaire, and the information sheet.

Stage 5 - **Analysis of Questionnaires (First Round):** This round was conducted through two phases of data analysis and development of the feedback sheet. An analysis of the responses of the Likert scale voting and comments provided was performed and the recommended modifications by panellists were applied. Some processes were combined with other processes, moved to a different capability or removed completely from the model. The project management capability changed entirely and new processes were considered. This phase resulted in an improved version (Version 2) of the model.

Stage 6 – Data Gathering (Second Round): In the second round, the second open-ended questionnaire was forwarded to all nine panel members. Panellists were given two weeks to respond to the questionnaire. It took some panellists three weeks to reply.

In this round, a feedback sheet summarising the analysis of data gathered in the first round was forwarded to each panellist. The main purpose of feedback is to give each panel member a chance to revise their original judgement about the model in light of the judgement of others.

The feedback sheet provided summaries of participants' comments from the first round on the entire model, capabilities, processes and sub processes and clarified certain misconceptions about the model. These were supported by a bar chart showing the total number of votes on the 5-point Likert scale (0-4) for each assessed capability and process (see Appendix C).

Step 7- Analysis of S Questionnaires (Second Round): The results of analysis in the second round showed strong agreement between panellists on the rating of processes and sub processes (see Appendix C). Initially three Delphi rounds were planned, but since a satisfactory degree of consensus was reached at the end of the second round, a decision was taken not to proceed to a third round. The majority of sub processes gained consensus; however, the analysis revealed very slight disagreement regarding a very few sub processes. Chapters 5 and 6 provide a full discussion of data analysis and findings.

4.5.7 Participant Anonymity

The Delphi method was designed with appropriate strategies to ensure that participants' anonymity and confidentiality were maintained. The Delphi questionnaires did not contain any personal identifiers, instead identification numbers were used to identify panel members and these were used in both rounds.

4.5.8 Panel Structure and Size

There is no consensus in the literature on what constitutes an ideal number of panel members in a Delphi study. As reported by) Hallowell and Gambatese (2009) empirical studies showed no correlation between the size of the panel and the accuracy and effectiveness of the study. The size is variable depending on the problem under study, time and cost constraints. However, it has been suggested that in a heterogonous group, as in this study, 10-15 is enough (Hanafin & Brooks, 2005). Researchers of Delphi studies asserts that the Delphi panel size is dependent on group dynamics to reach a consensus, and useful outcomes can be attained from a small panel size. It has been reported by Day & Bobeva (2005) that seven is a suitable panel size. This is a very important consideration because in Delphi studies it is expected the size of participants to drop after each round.

To achieve variation in panellists experience and background, the sample respondents of this study included three different geographical regions including North America, Australasia and South Asia. Moreover, the collective experiences of panellists cover the five capabilities. Five of the panellists have relevant experience in SSEs.

Table 4-3 lists the demographic information about panel members.

Table 4-3 : Demographic Information about Participants

		# of Participants
2. GEOGRAPHIC LOCATION	North America	3
	South Asia	2
	Australia and New Zealand	4
	Total	***9***
3. NUMBER OF EMPLOYEES	1-10	1
	10-30	1
	30-50	1
	>50	3
	Unidentified	3
	Total	***9***
4. YEARS OF EXPERIENCE	5-10 years	2
	10-25 years	3
	25-50 years	4
	Total	***9***
5. ROLES ASSUMED	Software Process Improvement	4
	Software Quality Audits	4
	Quality Assurance	5
	Project Management	4
	System Analyst	2
	Solution Architect	4
	Software Development	4
	Software Testing	2
	Release and Deployment	2
	Configuration Management	1
	Software Maintenance	1
	Others	4
6. CERTIFIED SPI CONSULTANTS	Holders of SPI Certifications	5

7. QUALIFICATION	Undergraduate	5
	Post Graduate	4
	Total	
8. MONETARY VALUE OF PROJECTS	Less than $50,000	
	$50,000 – $100,000	1
	$100,000-$500,000	4
	$500,000-$1000,000	3
	more than $1000,000	1
	Total	9
9. EXPERIENCE IN SMALL SSE	Yes	5

4.5.9 Survey Instruments

The study was completed in two rounds, during which questionnaires with open ended questions were used to collect rich qualitative data in order to allow a wide array of views. Quantitative data was also collected by providing respondents with the opportunity to rate the processes using the 5-point Likert scale. The design of the questionnaires provided a chance for participants to present their comments and justifications for their answers. To enhance the validity, the questionnaire was forwarded to a postdoctoral researcher in information systems from North America for review and feedback. The feedback came back positive with minor thoughtful insights and wording to enhance and clarify instructions. The first round questionnaire included the initial version of the model, and demographic information was collected about participants.

The second round questionnaire included the revised model and a feedback sheet. No practices were dropped including those that reached consensus in the first round. This allowed panellists to reassess the entire revised model, which was the outcome of the first round. This is an accepted practice as reported by Von Der Gracht (2012).

As for the quantitative data analysis a five point Likert scale was used to evaluate two elements, namely importance and feasibility. Table 4-4 provides detailed definitions of the scoring system that was provided to participants in both rounds to ensure common understanding among them. In essence, importance relates to the relevance of the practice to SSEs and to determine its inclusion in the model. Feasibility relates to the affordability of the enterprise to implement and manage the practice. Because the questions were designed to elicit opinion on whether the proposed practice should be included or removed from the model as well as justification of opinion, the scoring system did not include a score of indeterminate importance or indeterminate desirability.

Table 4-4: Definition of Importance and Feasibility

(4) Very highly important	Highest relevance to the model, highest impact on the Organisation, first priority.	(4) Very highly feasible	Can definitely be implemented, no increase in available resources needed, acceptable to all in the Organisation.
(3) Very important	High relevance to the model, high measurable impact on Organisation, Second priority.	(3) Highly feasible	Can be implemented, no major increase in available resources needed, acceptable to majority in the organisation.
(2) Important	Relevant to the model, has a measurable impact on the Organisation, low priority.	(2) Feasible	Can be implemented, slight increase in available resources needed, acceptable to majority in the Organisation.
(1) Low importance	Very low relevance to the model very low measurable impact on the Organisation, no priority.	(1) Not very feasible	Some indication that it cannot be implemented, major increase in available resources needed, unacceptable to majority in the Organisation.
(0) Not needed, (remove)	The practice has no relevance and can be removed from the model.	(0) Not feasible (remove)	Cannot be implemented, extremely large scale increase in available resources needed, completely unacceptable to all in the Organisation.

4.5.10 Consensus Measurement

Delphi technique is characterised as a systematic process for reaching an informed group-based decision, therefore, consensus measurement is very important for evaluating agreement among panellists, and is a determinant measure for ending further iterations. There is no general standard for measuring consensus in Delphi studies and it varies from one study to another (Rayens & Hahn, 2000; Von Der Gracht, 2012).

Descriptive statistics such as measurement of central tendency (i.e. mean, median and mode) and dispersion (i.e. range, standard deviation and IQR) have been used in Delphi studies (Hasson & Keeney, 2011). Measures of central tendency have been analysed with measures of dispersion such as Interquartile range (IQR)(Von Der Gracht, 2012). IQR, which is a measure of dispersion, has an advantage that it is not sensitive to outliers and is very suitable for ordinal scales. On a 9-point ordinal scale, IQR of 2.0 or less represent consensus and 3.0 least agreement. On the other hand, when a four or five point scale is used, IQR of 0.0 represent definitive agreement, 1.0 and below is considered reasonable agreement and 2.0 is least agreement (Von Der Gracht, 2012). Additionally, frequency distribution has also been used by some researchers to assess consensus (Rayens & Hahn, 2000). Some researchers have also ended their Delphi rounds based on subjective judgement when additional rounds would not add to the results (Von Der Gracht, 2012).

Von Der Gracht (2012) and Rayens & Hahn (2000) reported examples of statistical measures used in different Delphi studies as presented in Table 4-5.

Table 4-5: Common approach to assess consensus (Von Der Gracht, 2012)

Measure	Example	Reported by
Subjective	Examples: The expert's rationale for response had to be consistent with the mean group response. The third round of the study would not add to the understanding provided by the first two; thus the study was stopped.	(Von Der Gracht, 2012)
Level of agreement	The percentage of votes that fall within a specified range, example: Consensus is achieved when 67% or more were positive (i.e. desirable or very desirable) or negative (not very desirable or not desirable).	(Von Der Gracht, 2012) (Rayens & Hahn, 2000)
Central tendency	Mode, Median, Mean Mean response is within an acceptable range (+/- 0.5) and with acceptable coefficient variation (50%)	(Von Der Gracht, 2012)

Measure	Example	Reported by
Variance	Standard deviation, Coefficient of variation Consensus is achieved when ratings for items fell within the range of +/- 1.64 standard deviation. Coefficient of variation at or below 0.5 indicate reasonable internal agreement Consistent decrease of Coefficient of variation between rounds is indicative of an increase in consensus.	(Von Der Gracht, 2012) (Von Der Gracht, 2012) (Von Der Gracht, 2012)
Interquartile range (IQR)	Interquartile range (IQR), IQR ranged from 0.00 (most agreement) to 3.0 (least agreement), examples: IQR of 2 or less on a 9 point scale IQR of 1 or less for 4, 5 or 7 point scale.	(Von Der Gracht, 2012) (Rayens & Hahn, 2000)

The literature pointed out that the mean is not a correct measure for Likert scale since it is very sensitive to extreme values; hence, variability should be calculated using interquartile range, but not standard deviation (Rayens & Hahn, 2000; Von Der Gracht, 2012). Median and mode can be considered instead of mean.

Additionally, the Delphi literature noted that consensus measurements should not be used as the sole criteria for stopping Delphi rounds (Dajani, Sincoff, & Talley, 1979; Von Der Gracht, 2012). Consensus alone may not reflect whether actual agreement has been reached, without the assessment of the consistency of participants' responses between successive rounds, which is referred to as group stability. Therefore, Delphi researchers regarded stability as a necessary criterion for stopping the rounds. The assessment of consensus is only considered once stable responses from participants have been obtained (Dajani et al., 1979).

In this study, to measure consensus, median and Inter-quartile range (IQR) were calculated for each practice. The median shows the middle score of what all panel members think about the practice, while IQR shows whether responses are clustered together, or scattered within the range. IQR of zero reflected definite consensus and IQR of 1.0 or less reflected strong agreement among panellists. On this basis and along with the considerations from the results of qualitative analysis, a decision was taken to consider a practice important or feasible. For this reason, bar charts rather than histograms were provided with the feedback sent to participants in a simple friendly manner so that they are aware of the group assessment of the practices.

As for measuring stability, the coefficient of variation was used as recommended by Dajani et al. (1979). The Coefficient of Variation (V) is calculated in each round and a decision was taken to continue or terminate the study when changes are within a predetermined stable range of values. A (V) value at or below 0.5 is used as a cut-off point for reaching stability in panellist responses between rounds (Dajani et al., 1979).

4.5.11 **Limitations of Delphi Technique**

The Delphi technique is criticised as lacking standardised methodological procedures and having shortfalls in the application of consensus measurement (Von Der Gracht, 2012), despite its widespread adoption in a variety of research domains. The literature reports that the implementation of the Delphi method lacked purposeful reflection and evaluation (Day & Bobeva, 2005). However, there are number of studies that presented toolkits and guidelines for successful application and management of Delphi (Day & Bobeva, 2005; Okoli & Pawlowski, 2004). Moreover, there are examples of successful applications of the Delphi technique as a research methodology (Hallowell & Gambatese, 2009).

This study adopted guidelines proposed by the literature to ensure rigour and reliability as suggested by Hasson & Keeney (2011). However, much work is still needed in this area particularly in the area of consensus measurement, rigour and analysis of results (Day & Bobeva, 2005).

4.5.12 **Attainment of Rigour**

The most important aspect of any research regardless of the design adopted is the attention given to rigour. The attainment of rigour in Delphi studies is dependent on the study design as each Delphi study has unique characteristics including the number of rounds and feedback.

The measurements of rigour differ between quantitative and qualitative approaches. Reliability and validity are the attained measures for quantitative research, while rigour is measured in qualitative research by applying trustworthiness, which relates to the effectiveness and appropriateness of a Delphi study (Hasson & Keeney, 2011). According to Hasson & Keeney (2011) the transfer of measurements between paradigms is problematic since interpretivists aim to understand situations while positivists seek predictions.

In this study, the epistemological stance is post-positivism with an interpretive framework employed to analyse qualitative data (J. Creswell, 2012). Therefore, the four criteria for trustworthiness identified in Guba's model have been applied to attain rigor (Hallowell & Gambatese, 2009). These four measures are discussed in Table 4-6.

Table 4-6: Four measures for the attainment of rigour in this study

1. Credibility Corresponds to internal validity in positivist research; the degree to which the results of the study can be relied upon i.e. a) the quality of the evidence revealed by participants, b) the validity of the research instrument and c) panellists' experience, and context.	Firstly, the inclusion of two iterations for data collection and analysis, which in its totality is more than the sum of the parts (Day & Bobeva, 2005) allows judging the appropriateness of panellists' judgement (Day & Bobeva, 2005; Shenton, 2004). Secondly, credibility is achieved by receiving feedback on the model by a sample of experts in the field (Hasson & Keeney, 2011). In this study and as per the recommendation of Day and Bobeva (2005), panellists were selected based on pre-defined selection criteria to ensure that participants collectively had a range of experiences covering all five capabilities. This is a form of triangulation, applied through data sources, which allowed the verification of viewpoints against each other (Shenton, 2004). Thirdly, an expert verified the data collection instruments and full considerations were given to the organisation of the questionnaires and clarity of questions. In the first round, participants made a few

	wrong assumptions about the model. These were clarified in the feedback sheet. Shenton (2004) suggests that random sampling of participants provides assurance that the sample is representative of the wider group. In this study, an announcement was published on LinkedIn forum and as a result, interested subjects participated in the study. The researcher has no relationship with any of the participants. Finally, Delphi naturally establishes a degree of credibility through its requirement for the anonymity of participants which ensures avoidance of group think and group bias.
2. Dependability	
Addressing the issue of reliability relates to the degree to which data can be believed and the in-depth coverage of the research design, methods used for data collection to ensure that proper research practices have been followed (Shenton, 2004).	As reported by (Shenton, 2004), researchers stress the relationship between credibility and dependability and that the demonstration of the former ensures the latter. In this study, guidelines suggested by the literature were adopted in specific areas of the research mainly: the selection of panellists, the methodology, and the administration of the rounds as well as the methods of assessing consensus. As recommended by the literature, the details of the process adopted in conducting this research inquiry were communicated here to enable future researchers to repeat the process (Shenton, 2004).
3. Conformability	
Corresponds to objectivity, conveys neutrality, which is related to the steps taken to ensure that the findings are the result of the collective decision	As well as providing justifications and references for the statistical methods used, strict rules were set and followed for making judgements on the analysed data. Panellists anonymity was ensured all through the study. The nature of Delphi study where panellists remain unknown to each other helped in avoiding

made by the panellists and are not influenced by the researcher's preferences.	group think or group bias. Moreover, researchers consider that a key criterion for conformability is providing justifications for the methods adopted and reporting the weakness of the employed technique (Shenton, 2004). In this study, a detailed description of the Delphi method, how data was collected, and findings formed is provided. All conformed to guidelines suggested by the literature as show in the previous sections of this chapter.
4. Transferability Corresponds to external validity and generalizability, and the extent to which the findings can be applied to other settings	The transferability and generalizability of the results are ensured through the multiple iteration of data collection and the feedback method, this in addition to the anonymity of respondents (Day & Bobeva, 2005; Hasson & Keeney, 2011). Additionally, site triangulation is achieved since participants belong to different organisations (Shenton, 2004). Further, interdisciplinary triangulation is utilised by including panellists from different discipline areas relevant to software engineering and SSEs. Finally, quantitative and qualitative methods of data collection are used to corroborate the findings.

4.6 Summary

This chapter has provided a detailed description of the ontological and epistemological stance undertaken, specifically post-positivism supported by quantitative and qualitative methods of data collection. This approach is found to be the best approach to support the research objectives and answer the research questions.

Moreover, the motivation for using the Delphi technique is clarified with a detailed description of the research process, the procedures undertaken to select panellists, and the process followed in conducting Delphi rounds. The chapter provides discussion and justifications for the quantitative methods used for data analysis and measuring consensus; these will be discussed further in the next chapter. Finally, considerations of rigour are addressed.

The next chapter presents the analysis of the data collected from the two rounds.

CHAPTER 5 - ANALYSIS AND FINDINGS

5.1 Introduction

This chapter builds on the previous chapter and presents the results of the research. It provides an analysis of the quantitative and qualitative data collected in the two-round Delphi study and a discussion of findings.

Following this introduction there is a discussion of the demographics of the participants in this research. The literature review on the Delphi method suggested the use of interquartile range (IQR) to determine the level of consensus. Therefore, to measure consensus, median and Inter-quartile ranges (IQR) were calculated for all the practices underlying the five capabilities. The justifications for these statistical measurements were discussed in detail in the previous chapter. However, this chapter provides an explanation of the analysis processes followed in both the rounds that constituted the consensus building steps for this study.

The two Delphi rounds fulfilled the primary research objectives, which is the improvement of a software capability framework or model by a group of experts and the development of the final accepted version of the model. Therefore, this chapter answers the following research questions:

RQ2. What are the specific capabilities critical for the desired model that would benefit SSEs at each phase of the software product lifecycle?

RQ3. What are the underlying processes desired for each identified capability suitable for SSEs?

RQ4. What are potential dynamic capabilities suitable for SSEs?

These are demonstrated in sections 5.7, 5.8 and 5.9, each discussing the results, analysis and findings arising from the two rounds for each of the five capabilities. The discussion of prominent findings from this research is presented in Chapter 6.

5.2 Demographics

Participants were asked questions concerning their roles, experiences and the type and size of projects implemented as presented in Table 4-3 in section 4.5.8. Analyses of these demographics were carried out as this influences the effectiveness and appropriateness of the findings of the Delphi study.

The members of the panel were involved in projects with varying investment sizes ranging from $50,000 and in some cases exceeding one million dollars. Five panellists have experience in small size enterprises and the roles assumed by all members cover the different activities and roles of the software development lifecycle. Four panellists are experienced software process improvement consultants and holders of certifications. Finally, they were recruited from the regions of North America, South Asia and Australasia. Since seven out of nine panel members had 10-25 years of experience that included software quality assurance, organisation audits, software process improvement, solution architects, software development and others, and five panellists had experience in SSEs. Therefore, the demographic information suggests that the panel members form a good representative sample for this study.

To keep the participants of this study anonymous, Table 5-1 provides a list of identifiers that will be used to reference their individual comments.

Table 5-1: Participant identifiers and general demographic information

ID	Region	Experience (# of years)
3	South Asia	25-50
4	North America	10-25
7	Australia and New Zealand	5-10
9	Australia and New Zealand	5-10
10	Australia and New Zealand	10-25
11	Australia and New Zealand	25-50
12	North America	25-50
13	South Asia	10-25
14	North America	25-50

5.3 The Analysis Process

Following are the steps followed for analysing the collected data:

- Summarising the qualitative comments obtained from panellists.
- Statistically analysing the collected quantitative data.

- Ranking the practices with respect to importance and desirability.
- Making judgements and taking a decision of whether to retain, modify or remove the practice.

5.4 Quantitative Data Analysis

As discussed in section 4.5.9, the panel members were asked to rate the importance and feasibility of the capabilities and their suggested processes on a five point Likert scale (0-4). The scale did not include a neutral item to encourage participant to either approve or reject the practice. It is a known phenomenon and referred to as central tendency error, that in controversial topics participants tend to avoid extreme opinions believing that a neutral option is a safer choice.

The participants were provided with clear explanations of the different items on the Likert scale to avoid multiple interpretations of the items, which would therefore lead to unreliable findings (Refer to Table 4-4 on page 88).

The data from each round were analysed in an Excel spread sheet using measures of central tendency and dispersion. Feedback to participants included percentages of panellists' votes on the items on the Likert point scale. These were provided in the second round along with summaries of panellists' comments, and were also used in the analysis of results. The central tendency and dispersion calculations included mode, median, and interquartile range (IQR) scores for all the processes and sub-processes in the model. As discussed in section 4.5.10, IQR was used to assess panellists consensus, a small interquartile range (IQR) is an indicator of high consensus that is responses that are closely clustered on one or two items of the Likert point scale, and more than 50% of opinions fall within 1 point on the scale.

The mean and standard deviation are not robust measurements for ordinal data since they can be influenced by extreme values as discussed in the previous chapter. To illustrate using an example, in Table 5-2 process (1.2.1) shows high consensus for importance (IQR = 0.5 and MOD = 3.0) despite a mean score of 2.6. This is because five participants considered it highly important (Likert Item =3), two participants rated it very highly (Likert Item =4) and two participants rated it low (Likert item =1). Comparably, feasibility results for process (1.2.2) had a slightly higher mean (2.9) but showed very low consensus (IQR = 2.25), this is because four participants rated it low (Likert item 1) and one selected (Likert item =2). We can clearly understand from this example that median is a more appropriate measure for ordinal data when extreme values are possible. The description and justifications of the data analysis measurements used in this study were covered in detail in the previous chapter.

Table 5-2: Quantitative analysis example

			Importance (0-4)						Feasibility (0-4)				
1.2	**PROCESS REVIEW**		1.2.1	1.2.2	1.2.3	1.2.4	1.2.5		1.2.1	1.2.2	1.2.3	1.2.4	1.2.5
1.2.1	Plan Process Review	Mean	2.6	2.8	2.9	3.0	2.9	Mean	2.3	2.1	2.4	2.4	2.6
1.2.2	Collect and Analyse Process Data (DC – Sense)	Median	3.0	3.0	3.0	3.5	3.0	Median	2.0	1.0	2.0	2.0	3.0
1.2.3	Review Process Performance	Mode	3	3	4	4	4	Mode	2	1	3	2	3
1.2.4	Document Review Results	**IQR**	**0.5**	**0.75**	**1.5**	**1.5**	**1.5**	**IQR**	**1.25**	**2.25**	**1.25**	**1.5**	**1.5**
1.2.5	Report Review Results	STD	1.06	1.16	1.25	1.31	1.25	STD	1.04	1.36	1.06	1.19	1.19
		CV	0.40	0.42	0.43	0.44	0.43	CV	0.46	0.64	0.45	0.50	0.45
		% (4)	13%	25%	38%	50%	38%	% (4)	13%	25%	13%	25%	25%
		%(0)	0%	0%	0%	0%	0%	%(0)	0%	0%	0%	0%	0%

5.4.1 Criteria for Reaching Consensus and Stopping the Rounds

As detailed in chapter 4, the data collected from the questionnaires were keyed into Microsoft Excel spreadsheet and analysed. The results of the quantitative data were interpreted in reference to participants' comments. The Delphi method continued until the criteria for consensus or stability were met. The criteria for consensus are summarised in the Table 5-3.

Table 5-3: Criteria for measuring consensus

IQR= 0	Definitive consensus;
IQR (0 to 1)	Reasonable consensus;
IQR (1 to 2)	Poor degree of consensus; definite need for an additional round.

As for the stability of the voting, a coefficient of variation (CV) measure was used to assess group stability according to the following rules which were acquired from Dajani et al. (1979).

Table 5-4: Coefficient of variation decision rule

CV= (0 to 0.5)	Good degree of stability; no need for an additional round
CV= (0.5 to 0.8)	Less than satisfactory degree of stability; possible need for an additional round
CV >= 0.8	Poor degree of stability; definite need for an additional round

The percentages of median scores of very highly important %(4) for each sub process have been reported, and thus, higher scores reflect participant agreement on the importance and feasibility of the practice. Moreover, the percentages of median scores for removing the sub process (Likert item = zero) or %(0) have been reported to reflect the agreement on excluding the practice from the model.

The same analysis procedure used for the first round was followed in the second round. However, panellists were asked to categorise each practice in the model into either basic or advanced practice. For data analysis the value (1) was used for basic and the value (2) for advanced. IQR and MOD were calculated to evaluate panellists' decision as shown in Table 5-5.

Table 5-5: Quantitative analysis of classifications

	Classification (A/B)					
	1.3.2	**1.3.3**	**1.3.4**	**1.3.5**	**1.3.6**	**1.3.7**
B(%)	43%	43%	43%	43%	43%	43%
A(%)	57%	57%	57%	57%	57%	57%
Mode	2	2	2	2	2	2
IQR	**1**	**1**	**1**	**1**	**1**	**1**
	A	**A**	**A**	**A**	**A**	**A**

5.5 Supporting Evidence and Provided Information

The complete set of results analysis for each capability is provided in Appendix D. The feedback sheet provided to panellists in the second round (see Appendix C) includes panellists' comments on the model provided in the first round. In the second round few quantitative data were collected, these were presented in the discussion of results. The first round questionnaire in Appendix C includes the first version of the model. The description of sub processes for the first version of the model is provided in the information sheet (see Appendix C). The second round questionnaire provided in Appendix C includes the second and final version of the model along with the descriptions of sub processes

5.6 International Standards

One question relating to international standards has been added to the first round questionnaire asking panellists about whether they consider the adoption of standards to be adequate and feasible for SSEs. Existing standards are accepted by the literature as being best practice by being rigorous and including important processes (A Cater-Steel, 2004a). However, there is a concern by the software community that existing international standards are not suitable for SSEs because they require huge resources (Pino et al., 2008, p. 248). The objective of this question is to gain insights on panellists' perceived value of adopting international standards in SSEs.

The analysis of panellists' comments perceived that international standards may not be very suitable for SSEs. The comments provided also pointed out some weaknesses with standards, confirming reports.

Panellists expressed their concern that SPI could inhibit innovation and creativity. This view is discussed by Kelly and Culleton (1999, p. 41), where SSEs strengths are associated with the creativity and innovation of their employees, and international standards are viewed as leading to bureaucracy and inhibiting the freedom of individuals. This view is presented in the comment below:

- *An international standard framework as a guideline for small software organizations may be helpful…, however, we have to be careful that the adoption of international standards will not slow down innovations and the need to react in an out of box manner to fulfil consumer needs (#11).*

For successful adoption of standards, panellists suggested that SSEs must receive external help to guide them in understanding the requirements of standards. One of the identified factors for enhancing the success of SPI initiatives is having access to experienced consultants or assessors. The use of experienced assessors or consultants as mentors for the implementation of SPI initiatives has been flagged by (A Cater-Steel et al., 2006). Following is a comment that conveys this view provided by one panellist:

- *I do not believe that international standard would help. SSEs need to be flexible and adopt a new approach quickly to survive. International standards tend to be complex and written in a way that is difficult to understand at times requiring small organisations to seek assistance in interpretation which could present a barrier to adoption (#4).*

Additionally panellists expressed concern that SSEs' motivation for adopting standards is normally to gain certifications rather than improving processes. This agrees with the findings of Pino et al. (2008, p. 248) who found that in most cases improvement efforts in SSEs do not lead to gaining certifications. The following panellist comment expresses this view:

- *Most companies ignore international (and other) standards, with the exception of a few that use CMMI, ISO9001 and PMI. When they use international standards they usually do it for one of the following reasons:*
    - *CMMI motivation: appraisal rating to sell to government agencies*
    - *ISO motivation: selling to Europe*
    - *PMI motivation: Trying to look as if they do project management*

  The motivation should be to improve performance rather than receiving certifications. There are always exceptions, and in a few cases the do adopt these frameworks properly (#14).

5.7 Dynamic Capabilities

This section provides analysis related to the fourth research question: (R4) 'what *are potential dynamic capabilities suitable for SSEs?'* Section 5.9 provides analysis at the process and sub process level, including analysis and justifications of the proposed DCs for each capability. Therefore, this section and section 5.9 both contribute to answering RQ4.

The concept of dynamic capabilities is essentially to ensure enterprise agility and survival as markets and technologies change. A number of practices on the proposed model were selected as DCs. There are four types of DC activities used.

- **Sense** to identify opportunities and threats by exploring technological opportunities, scanning and evaluating the markets, listening to customers, as well as scanning the other elements of the business environment and quickly carrying out internal and external transformation ahead of the competition.
- **Seize** by mobilization of resources to address the opportunities.
- **Transform**: to maintain competitiveness through continual improvement.
- **Enabler**: is an important capability or process to establish dynamic capabilities.

Each identified DC was categorized as either sense, seize or transform capability or received multiple roles depending on the tasks assigned to it .

The feedback of panellists was very positive regarding the concept of DCs on the model. Figure 5-1 shows that seven out of nine panellists considered the adoption of DCs important, eight considered them feasible for application in SSEs. Despite this, a few (N=2) panellists were a bit reserved about DCs and considered them as not very critical for the success of the organisation.

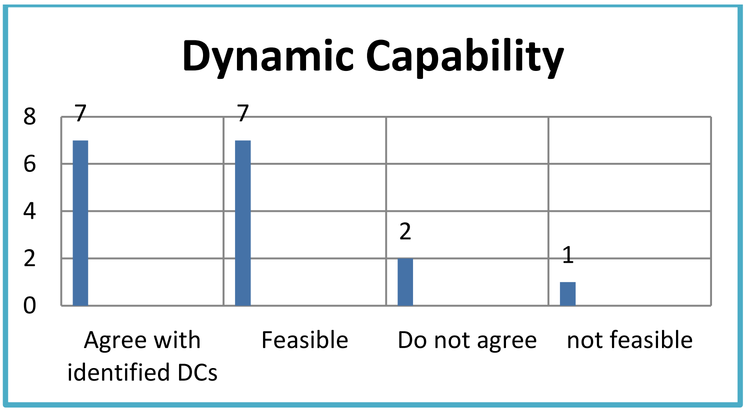


Figure 5-1: PD&I Dynamic Capability results

As conveyed through the comments provided below, some panel members understood the aspired benefits of empowering the model with DCs and suggested methods to make them feasible for adoption in SSEs. For a full list of comments concerning DCs, refer to the feedback sheet distributed to panellists in the second round (see Appendix C).

- *I think Dynamic Capabilities are just as important for SSE as they are for larger organizations, and critical in today's highly volatile world. The processes and the roles involved in supporting the DCs need not be as complex. SWOT analysis must be adopted for the SSE to understand and respond to current strengths, weaknesses, opportunities and threats (#11).*
- *The accountability for assessing and maturing DCs can be owned by a role or embedded in an annual or semi-annual planning process that includes key stakeholders from the business, the senior management group and IT (#9).*
- *The need for DCs depends on the market dynamism; in highly competitive markets the DCs approach will help. However, there is a need to assess benefits against cost (#3).*
- *I agree that the dynamic capabilities are generally relevant in a software development process but not necessarily crucial to enable success for SSE (#4.)*

Additionally, one member has totally rejected this concept, justifying that it is very academic and would add an unneeded level of complexity for SSEs.

- *The Dynamic Capability approach adds an unneeded level of complexity; it adds another level of academia (#14).*

Because the concepts of DCs are not common among the software development community, panellists conveyed their acceptance of DCs and they recommended minor changes.

In the next section, panellists' ratings of the five capabilities are analysed and discussed, including a discussion about the identified DCs for each capability.

5.8 Evaluation of Proposed Five Capabilities

To attain a holistic approach, five capabilities were identified that are regarded as important for attaining holistic SPI initiative in SSEs. The five capabilities provided a structure for driving the model and were also used as taxonomy when comparing the international standards. The five capabilities are presented in Figure 5-2.

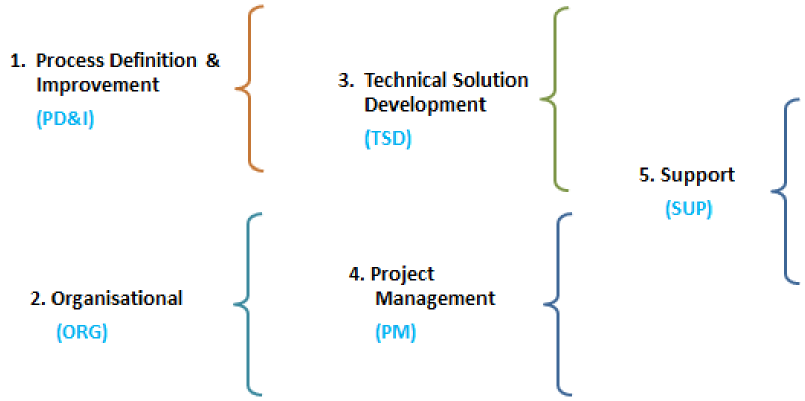


Figure 5-2: Proposed Software Process Capabilities

This section attempts to answer the second research question, (RQ2) '*What are the specific capabilities critical for the desired model that would benefit SSEs at each phase of the software product lifecycle?*

Hence, prior to evaluating the processes and sub processes, panellists were asked to rate the importance and feasibility of the five capabilities to the SSEs. To achieve this, a section was added to the first round questionnaire and panellists were asked to rate the importance and feasibility of each capability using a Likert scale 0-4, along with adding open-ended comments on each capability.

The analysis of results show absolute agreement among panellists on the importance of all five capabilities as depicted in Figure 5-3. The majority (N=>65%) of panellists considered the proposed capabilities as very highly important, except for ORG capability (N=44%). In regards to feasibility, the majority considered TSD and SUP to be very highly feasible (N>=67%), followed by PM (N=44%), and PD&I and ORG. (N=33%).

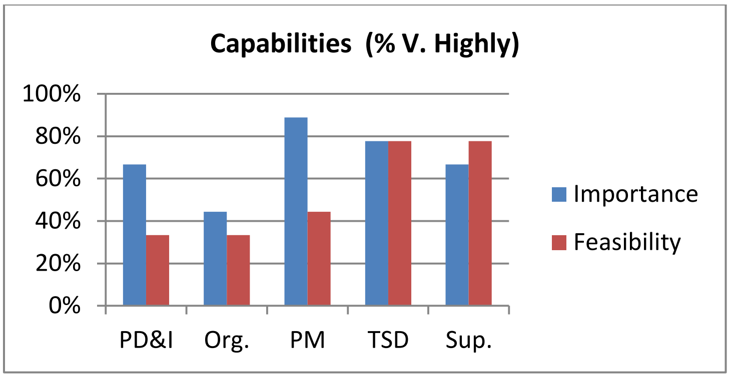


Figure 5-3: Panellists votes on the five capabilities (%Rating 4)

Despite the majority of panellists rating all five capabilities very highly (as can be shown in Table 5-6) results reveal a lack of consensus (IQR=2) regarding both importance and feasibility for PD&I. Meanwhile, ORG results show a lack of consensus regarding feasibility only.

Table 5-6: Panellists ranking of the five capabilities

		Importance	Feasibility
1. Process Definition & Measurement (PD&I)	Median	4	3
	IQR	*2*	*2*
	% v. highly	*67%*	*33%*
	% +ve	*89%*	*78%*
2. Organisational (ORG)	Median	3	3
	IQR	*1*	*2*
	% v. highly	*44%*	*33%*
	% +ve	*100%*	*100%*
3. Technical Solution Development (TSD)	Median	4	4
	IQR	*0*	*0*
	% v. highly	*78%*	*78%*
	% +ve	*100%*	*89%*
4. Project Management (PM)	Median	4	3
	IQR	*0*	*1*
	% v. highly	*89%*	*44%*
	% +ve	*100%*	*89%*
5. Support (SUP)	Median	4	4
	IQR	*0.25*	*0.5*
	% v. highly	*67%*	*78%*
	% +ve	*100%*	*100%*

These findings were justified by panellists through their qualitative comments as presented. Panellists stressed that SSEs must prioritise the adoption of the five capabilities based on the firm's maturity level and available resources. They stressed that the capabilities that should receive highest priority by SSEs are those that are core for the development of the product. Some panellists referred to these as bread and butter. Hence, TSD and SUP received the highest ratings followed by PM and the lowest were PD&I and ORG.

- *For a holistic approach, all capabilities are important; the organisation has to calibrate its focus based on needs/gaps (#7).*
- *The degree to which each is executed should be adapted to meet organisation's and product goals (#9).*
- *I think in an ideal world, we like to think all processes are feasible for a SSE but in reality they usually operate with limited resources (#12).*

Below are selected comments on each capability:

1.	***PD&I***	*"I see Process Definition and improvement to have the lowest priority for SSE" (#14).* *"It is an extremely important capability, but normally not handled well by organisations" (#10).* *"It can be a challenge for SSEs to dedicate time and resources for this capability however it is feasible for them to do so" (#4).*
2.	***ORG***	*"Requires leadership and is less likely to be handled well" (#14).*
3.	***TSD***	*"Very important, has direct impact on clients, often demanded by client" (#9).*
4.	***PM***	*Most essential set of practices might not be seen as essential by SSEs (#12).*
5.	***SUPP***	*"Very important, has direct impact on clients, often demanded by clients" (#9).*

It is worth commenting here that the lower rating concerning the feasibility of PD&I and ORG came from those panellists involved in SPI programmes rather than programmers or analysts because the first group are more focused on SPI challenges faced by SSE, while software engineers felt the need for them. This fact proves the importance of the inclusion of PD&I capability into the model and the reason why it is not given high priority by organisations could be the lack of adequate financial and human resources. In chapter 2, it was highlighted by Alexandre et al. (2006) that the lack of resources assigned to software development activities was a key characteristic of SSEs and the main barrier in their implantation of SPI programmes. Additionally, reviewed statistics in chapter 2 revealed that SSEs are characterised by having high employee turnover and a high number of part time contractors (Ministry of Economic Development, 2010). Both characteristics can disrupt the stability of the work environment resulting in negative impact on productivity and work efficiency. Additionally, SSEs are highly dependent on the experience of employees within the organisation. Often there is no proper documentation of organisational processes and once the individuals leave, the organisation suffers. This highlights the need for the institutionalisation of organisational processes and work procedures, as it is likely to stimulate organisational learning and growth. However, there is a need for future empirical research to assess this association.

5.9 Evaluation of Processes and sub Processes

The previous section examined the five capabilities as a whole, without disaggregation into processes and sub processes, providing panellists with an opportunity to assess the model at a high level. This section will discuss each capability in detail, providing quantitative and qualitative analysis of panellists' responses on the underlying processes and sub processes as well as a discussion of some conclusions. Therefore, the section seeks to answer the third and fourth research questions: (RQ3) *What are the underlying processes desired for each identified capability suitable for SSEs?,* and (RQ4) *What are potential dynamic capabilities suitable for SSEs?* It is worth to mentioning that RQ4 was also discussed in section 5.7.

5.9.1 Process Definition and Improvement (PD&I)

1. Process Definition & Improvement (PD&I)

- **1.1 Process Establishment**
- **1.2 Process Review**
- **1.4 Process Improvement**

5.9.1.1 *Analysis of Results*

PD&I capability consists of three processes: *1.1 Process Establishment*, *1.2 Process Review* and *1.3 Process Improvement*. The processes and sub processes with the complete results sets for PD&I in the first and second round are provided in Appendix D.

The first round results for *Importance* show an apparent lack of consensus for 1.1 (IQR=2), and 1.2 (IQR <=1.5), while 1.3 showed consensus for all sub-processes (IQR=1). Despite that, all results show a coefficient of variance (CV) lower than 0.5, which triggers the need for another round. Results are presented in Figure 5-4 below for the three processes in the PD&I capability; the full set of results is provided in Appendix D.

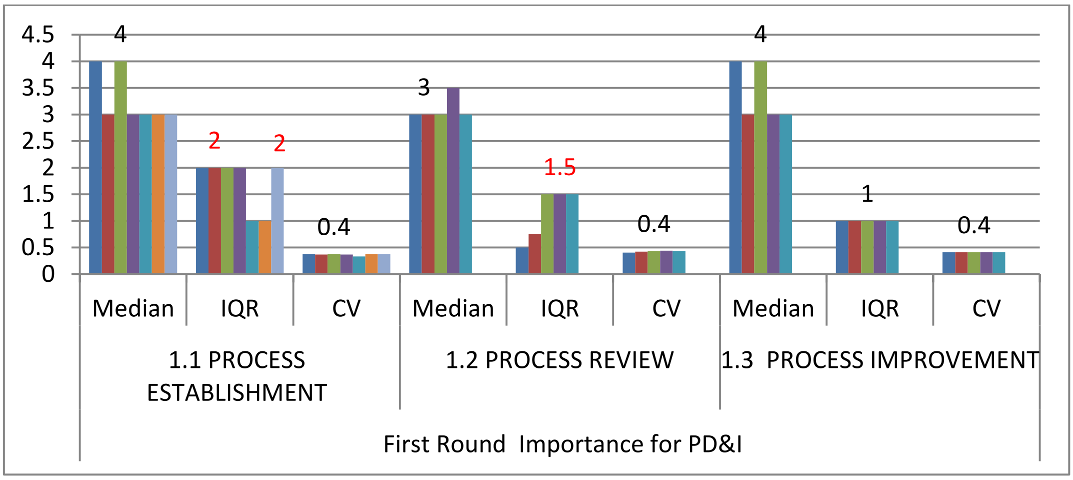


Figure 5-4: First round PD&I results for IMPORTANCE

The high IQR shows an apparent disagreement on the ratings, although none of the sub processes had a median less than (3), except for the sub process 1*.3.6 Update organisation's process asset library* (MED=2). As can be deduced from the collected qualitative data and the excerpts presented below, respondents consider this capability very important and crucial for SSEs to measure and improve their processes and to customise them to meet the arising business needs. Moreover, panellists are also convinced that the adoption of this capability enables delivering consistent product quality and promotes organisational growth.

- *It is critical for SSEs to describe and improve processes if they are to expand and grow without losing control of quality and reputation (#7).*
- *Process definition is critical for organisational performance. However, SSE must establish processes that work for them and avoid being swamped with large cumbersome processes. In my dealings with SSEs the target is always to document a process in one page(#11).*
- *It is important to have established processes to standardise the outputs of a software product/project, however, it is hard for SSEs because of resources and time constraints (#12).*

However, in the first round one panel member expressed the view that SSEs must give this capability low priority, and should focus on the core processes that generate revenue such as those of a technical nature. The findings of the literature review in chapter 2 revealed that this view is common among members of the software development community in SSEs and is supported by empirical studies as reported by A Cater-Steel (2004a, p. 246) in her study of 22 SSEs. Based on similar findings established during the literature review in chapter 2, this study supports the argument that a holistic approach that focuses on the entire system within the organisation is needed in SSEs rather than only the technical engineering aspect.

Following is an example of a panellist comment supporting this view:

- *A software team doesn't need this initially, compared with the engineering and management practices, therefore it is given lower priority (#14).*

As presented in Figure 5-5 below (full results in Appendix-E), the first round results for feasibility reveal a lack of consensus concerning most of the sub processes for 1.1 (IQR =2) and 1.2 (IQR >=1.5), only one process for 1.3 (IQR=2). It also reveals a lower median score (MED = 2-3) and slightly higher variability (CV >= 0.5) for the three processes. The lower median indicate that this capability is not perceived as very feasible for SSEs and the higher CV suggests the need for another round. In reports from studies assessing the capabilities of software processes in SSEs, A Cater-Steel et al. (2006, p. 256) reported consistent findings where eight processes were assessed in the 22 SSEs, process establishment was found to have the lowest capability in most assessed firms compared to the other processes. The assessed processes spanned the capabilities of PD&I, TSD, PM and SUP defined by this study.

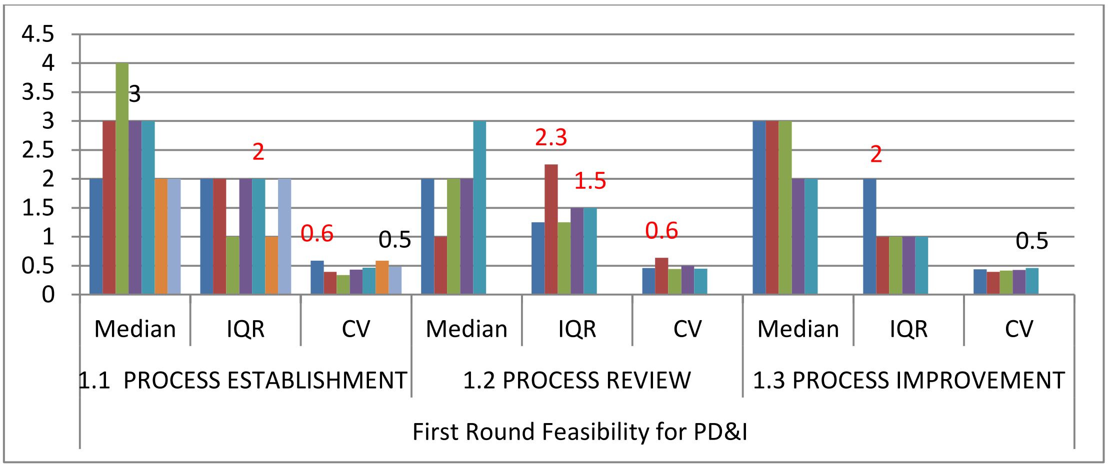


Figure 5-5: First round PD&I results for FEASIBILITY

The qualitative data supports the quantitative results, confirming what is reported by the literature that process improvement initiatives are very demanding and challenging for SSEs given their inherent human and resource constraints (Fening et al., 2008). Therefore, some panellists have expressed the need for SSEs to start improving those processes that are most critical for their business. The following comments imply this message:

> *I think this capability is important but is often regarded as an overhead or overkill in many small organizations (#13).*

> *The key is to identify those processes that are critical in establishing solid practices to sustain the organization over time and that can deliver meaningful insights for continuous improvement (#9).*

For a holistic approach, all capabilities are important. However, the organisation has to calibrate its focus based on needs/gaps (#12).

The analysis of panellists' comments reveals that some panellists emphasised the need for light processes to improve the feasibility of this capability.

- *It is a critical for organisational performance. However, SSE must establish processes that work for them and avoid getting swamped by large cumbersome processes. In my dealings with SSEs the target is always to put "process on a page" (#11).*
- *Processes for this capability must be simple, lean and compatible with agile development, which is very common methodology used by small or any organisation. Process definitions should be 1-2 pages. I would stress that you mean "LITE Process Definition and Improvement". Should be 'light pragmatic' processes, not complex processes (#14).*

These recommendations conform to the position taken by this study that SPI initiatives must meet the organisation's priorities and available resources. Therefore, this study's proposition stresses that SSEs must initially plan to improve processes from the different capabilities gradually. This can be achieved by selecting focus areas to start with, and once these are improved they can plan to improve another set. Through this gradual improvement and deployment of the model, the organisation's ability to deploy the entire model will be enhanced over time. Hence, this proposition should enhance the feasibility of this capability and the entire model.

In the second round (refer to the analysis of the quantitative results in Appendix D), results show a drop in IQR indicating higher agreement on all the sub processes for ratings of Importance (IQR <=1), except *1.3.2 Planning Improvements,* which shows slight disagreement (IQR=1.5) as shown in Figure 5-6. CV readings have dropped below 0.5 for all the processes, which is an indicator of stability in responses. Moreover, as can be seen from the results of the second round (refer Appendix D), there is an increase in panellists' rating (4) for all the sub processes. The improvement in ratings is driven by the feedback sheet provided in the second round, which is seen to raise awareness of the importance of this capability.

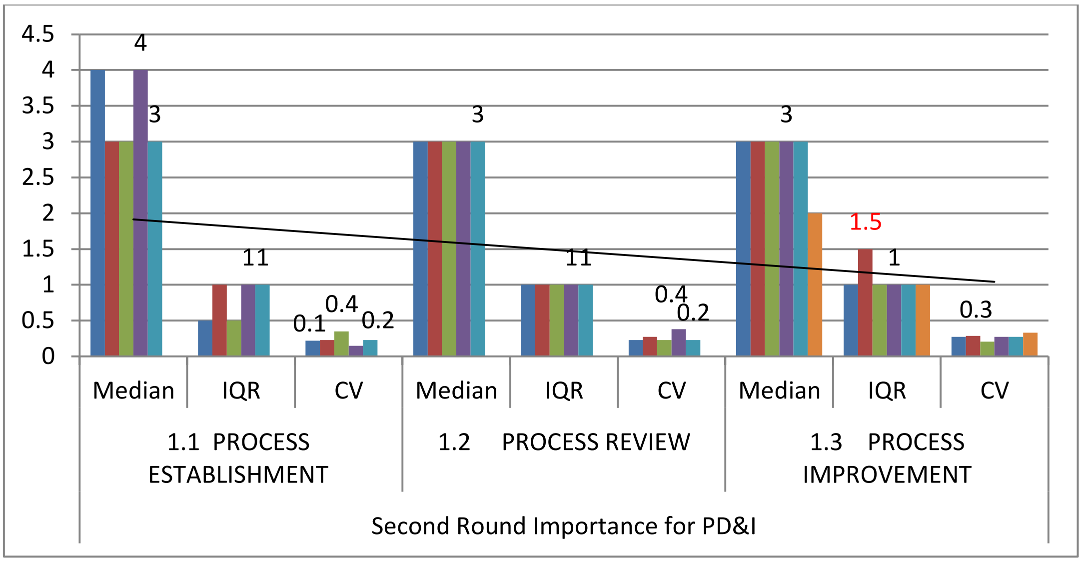


Figure 5-6: Second round PD&I results for IMPORTANCE

The analysis of feasibility results in the second round reveals apparent consensus on all the sub processes for importance, except *1.1.5 Define process measurements* as well as most of the sub processes of *1.2 Process Review* (IQR =1.5), in addition to *1.3.1 Identify and prioritise* process improvements and *1.3.2 Plan improvements*. This is presented in Figure 5-7 and the analysis of the quantitative results are provided in Appendix D.

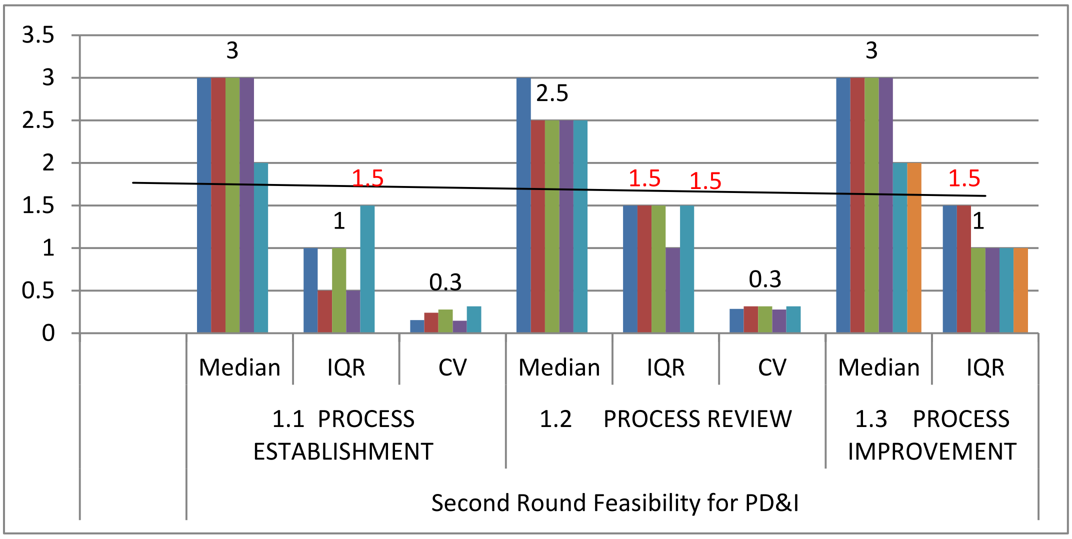


Figure 5-7: Second round PD&I results for FEASIBILITY

The analysis of panellists' comments on the sub processes reflects the low priority given to setting process measurements and conducting process reviews and improvements as reiterated by panellists' comments presented below. This view agrees with the findings of Wilkie et al. (2005) who assessed the adoption of CMMI practices in six SSEs; the findings of their study found poor adoption of measurements and analysis practices among the six organisations they studied, and found that measurements are only restricted for the evaluation of work products and services. Their findings explain that SSEs failure in measuring their processes reduces their ability to identify non-compliance issues or perform continuous service improvements. This justifies why the results in round 2 show low feasibility ratings for reviews and improvements, as panellists expect that SSEs are unlikely to collect process data; consequently, process reviews and improvements cannot happen without the existence of meaningful and effective measurements. Panellists convey this through their comments as presented below:

- *It is very important to define measurements for continuous process improvement, something unfortunately SSEs tend to neglect (#12).*
- *It is important to have measurements to measure performance. However, it is not feasible to collect measures for all processes and for every project especially when some of the measures are not numeric/intangible (#9).*
- *It is Important to review and improve but not very feasible due to time and budget constraints (#4).*
- *Planning Improvements is not feasible in SSEs given the limited resources and competing priorities. For practical reasons, it is something that could be done annually (#11)*

5.9.1.2 *Dynamic Capabilities*

Process Establishment is responsible for the creation of an organisation's processes, both operational and DCs. One important outcome of this process is the development of key performance indicators (KPIs) and measurements to assess the performance of processes. These measurements enable the enterprise to refine and upgrade their processes achieving continuous process improvement.

As discussed in the literature review, DC processes are instruments for improving operational processes as well as themselves in response to changes in the market place and the internal enterprise environment (Eisenhardt, 2000; Helfat et al., 2009; Zollo & Winter, 1999). DCs require more attention to refinements than operational processes to constantly assess the value of the competence they support in a high changing business environment (Teece & Pisano, 1994; Zollo & Winter, 1999). Therefore, *1.1* is classified as a DC Enabler, a prerequisite for dynamic capabilities.

In the first version of the model, a number of DCs were identified for *1.2 Process Review* and *1.3 Process Improvement*. In the second round, the model was reviewed and DCs were consequently refined as shown in the following section.

Seven out of nine panellists accepted the identified DCs and only one panel member commented suggesting the following change. The comment was not considered as the DCs are selected based on the activities and outcomes of each sub process.

- *I suggest that all the activities under Process Review are DC Sense. All the activities under Process Improvement are DC Transform (#12).*

5.9.1.3 *Classification of Sub Processes*

As shown in Table 5-7, the results for the classification of sub processes show a consensus for the thee PD&I processes. Panellists collectively classified all *1.1 Process Establishments* as basic practice. This reveals the importance given to the adoption of process driven organisational practices. However, panellists classified *1.2 Process Review* and *1.3 Process Improvement* as advanced practices. This conforms to what was reported by the SPI literature as presented in the above discussion.

5.9.1.4 *Final Changes to the model*

The second version of the model included moderate changes to processes 1.1 and 1.3. A comparison between version 1 and version 2 is shown in Table 5-7. The details of changes are presented in the feedback sheet provided to panellists in the second round (Appendix C). 'B' refers to basic and 'A' to advanced practice.

Table 5-7: Transformation of the PD&I Capability from Version 1 to Version 2

1.1 PROCESS ESTABLISHMENT		
	Dynamic Capabilities	**Classification**
Round 1	DC Enabler	
1.1.1 Initiate process establishment		
1.1.2 Define a process		
1.1.3 Identify work products		
1.1.4 Define architecture		
1.1.5 Define control mechanism		
1.1.6 Define process measurements		
1.1.7 Document a process		
Round 2	DC Enabler	
1.1.1 Initiate process establishment		B
1.1.2 Define a process		B
1.1.3 Define interfaces		B
1.1.4 Define control mechanism		B
1.1.5 Define process measurements		B
1.2 PROCESS REVIEW	`	
Round 1		
1.2.1 Plan process review		
1.2.2 Collect and analyse process data		
1.2.3 Review process performance	Sense	
1.2.4 Document review results	Seize	
1.2.5 Report Review Results		
Round 2		
1.2.1 Plan process review		A
1.2.2 Collect and analyse process data		A
1.2.3 Review process performance	Sense	A
1.2.4 Document review results	Seize	A
1.2.5 Report review results		A

1.3 PROCESS IMPROVEMENT

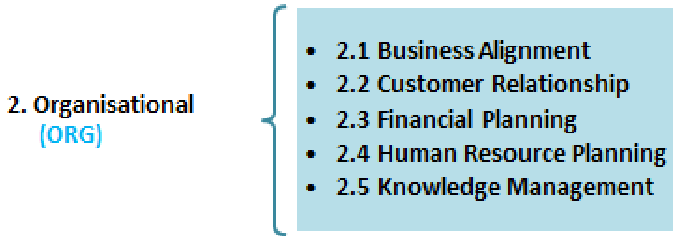
Round 1

1.3.1 Identify and prioritise process improvements	Sense, Seize
1.3.2 Plan improvements	Seize
1.3.3 Implement process action plans	Transform
1.3.4 Monitor and evaluate improvements	Sense, Seize
1.3.5 Update organisation process asset library	

Round 2

1.3.1 Identify and prioritise Process Improvements		A
1.3.2 Plan improvements	Seize	A
1.3.3 Develop improvement action plans	Seize	A
1.3.4 Implement process action plans	Transform	A
1.3.5 Monitor and evaluate improvements	Sense, Seize	A
1.3.6 Update organisation's process asset library.		A

5.9.2 Organisational Capability



5.9.2.1 *Analysis of Results*

The Organisational Capability (ORG) consists of five processes depicted in the diagram above. The complete set of results for this capability is provided in Appendix D.

As shown in Figure 5-8 and evident from the results provided in Appendix D, in the first round panellists expressed an agreement on importance (IQR <=1) for all ORG sub processes except for *2.5 Knowledge Management*. Interestingly, almost all the sub processes were rated very highly important by the majority (N=56% or more, MED =4), except for two sub processes related to *2.3 Financial Planning*. Although there was disagreement on the ratings of *2.3.1 Financial Planning* and *2.3.4 Invoicing* (IQR= 2), the remaining sub processes were rated highly important by the majority (N=56%, MED = 4). Following are excerpts from panellists comments for this capability:

- *No business large or small should operate without objectives and strategies. The deliverables of this process is the identification and prioritization of strategic goals (#11).*
- *Financial planning process is a DC Enabler: the better we understand our cost allocations and services, the more insight we have into potential savings / opportunities to improve services to make them more cost effective (#4).*
- *[Financial Planning] is basic and is required for all organizations but may vary in complexity depending on size and diversity of clients' portfolio, services and maturity of DC capability (#12).*
- *This process could be less formal in SSEs than large organisations, `and may take the form of monthly meetings between the business and IT managers and an annual report to coincide with the annual planning process (#14)*

Moreover, if we are to exclude process *2.5 Knowledge Management*, CV scores for the remaining sub processes show an acceptable level of stability (CV <0.5), which dismisses the need for a second round of data collection.

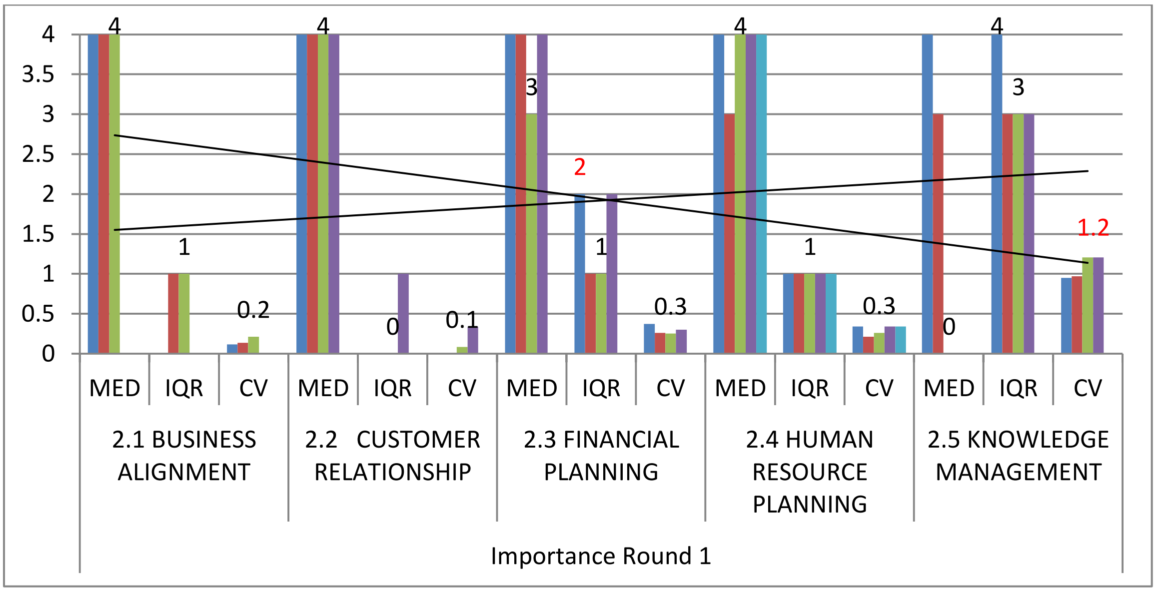


Figure 5-8: First round ORG results for IMPORTANCE

As shown by Figure 5-9 (full results are in Appendix D), similar positive agreement has been expressed by panellists towards feasibility for all ORG sub processes (IQR=1, MED = 3 or 4), with the exception of *2.1.3 Achieve strategic IT plans* (IQR =2, MED=3). Excluding 2.5 *Knowledge Managemen*t, results reveal good degree of stability in responses for feasibility (CV <0.5), which confirms there is less need for another round.

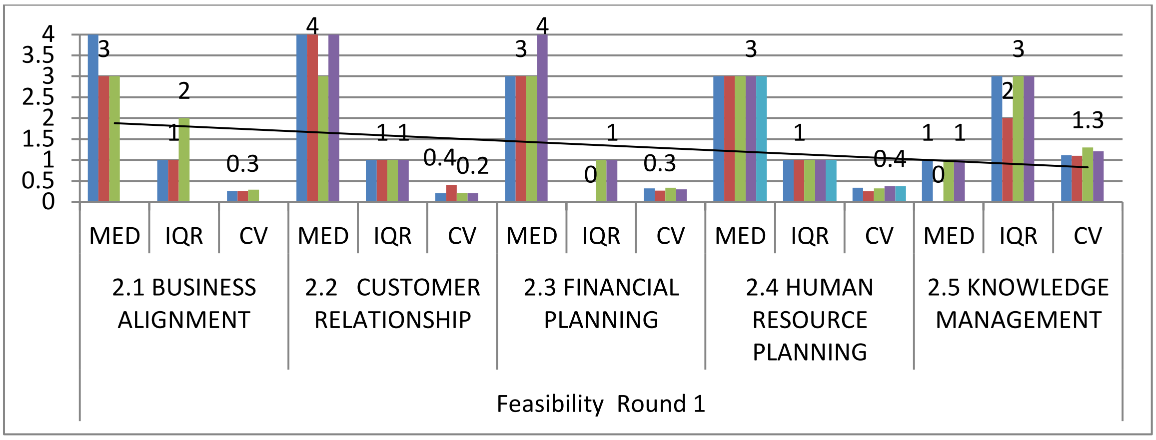


Figure 5-9: First round ORG results for FEASIBILITY

The analysis of results for 2.5 *Knowledge Managemen*t process shows very high disagreement (IQR =3 or 4), where panellists seem to be divided with regard to the importance of this process. Roughly, half the number rated this process highly or very highly, while the other half suggested its removal from the model. As for Feasibility, the majority (N=67%) agreed that this process is not feasible for SSEs (MED=0.0 or 1.0).

The analysis of quantitative comments presented below confirms panellists' strong support for this capability; this is interpreted as an evident growing awareness among members of the software community of the importance of organisational processes. The literature review in chapter 2 reported that SSEs focus on the technical engineering capabilities and do not consider organisational capabilities (A Cater-Steel, 2004a); the reported consequence of this approach is a lack of coherence between the internal and external environments (Dalcher, 2009). Moreover, a number of studies suggest that engineering processes function best when matched with other internal and external aspects of the organisation, such as human related factors, resources, culture, strategy and the market in which the organisation operates (Allison & Merali, 2007; Cattaneo et al., 2001; Kirk & MacDonell, 2009).

Following are excerpts of panellists comments expressing this view:

- *The organizational capability is fundamental for any software company that seeks to run a successful business. The degree and complexity of implementation may vary across organisations, but the key is that it must be implemented in a way that addresses the basic needs of the organization (#12).*
- *It is important for the company or business to establish clear [strategic] measurable goals, and it does not have to be complex. Therefore, it is feasible and necessary for accomplishing good software development (#4).*

  It is important and feasible through on-going engagement with customers. It needs to be managed and continuously improved (#9).

The qualitative and quantitative results agrees with the empirical findings of Fening et al. (2008), who reported that the organisation's performance will improve through the adoption of practices that tackle strategic planning, understanding of customers and market needs, and the implementation of human resource development and management.

In the second round, the knowledge management process was removed from the model. The ORG capability was included in the second round despite the consensus for both importance and feasibility and the stability in results. This decision was taken to allow for multiple iteration of data collection and feedback in order to enhance the transferability and generalizability of the results.

The results in the second round for both importance and feasibility are similar to the first round, with even more drop in CV readings for all sub processes as can be seen in Figure 5-10 and Figure 5-11. There is slight disagreement (IQR=1.5) regarding the feasibility of the sub processes for *2.1 Business Alignment* despite the fact that many respondents (roughly 43% or higher) rated them as very highly feasible (MED=4). Similarly, a high number of panellists rated this process very highly important in both rounds (N= 86% in round 2).

Furthermore, in round 2, a high percentage of panellists (N>70%) considered 2.2 *Customer Relationship Management* very highly important. Similar numbers were found for *feasibility* with a 12% - 30% increase from the first round.

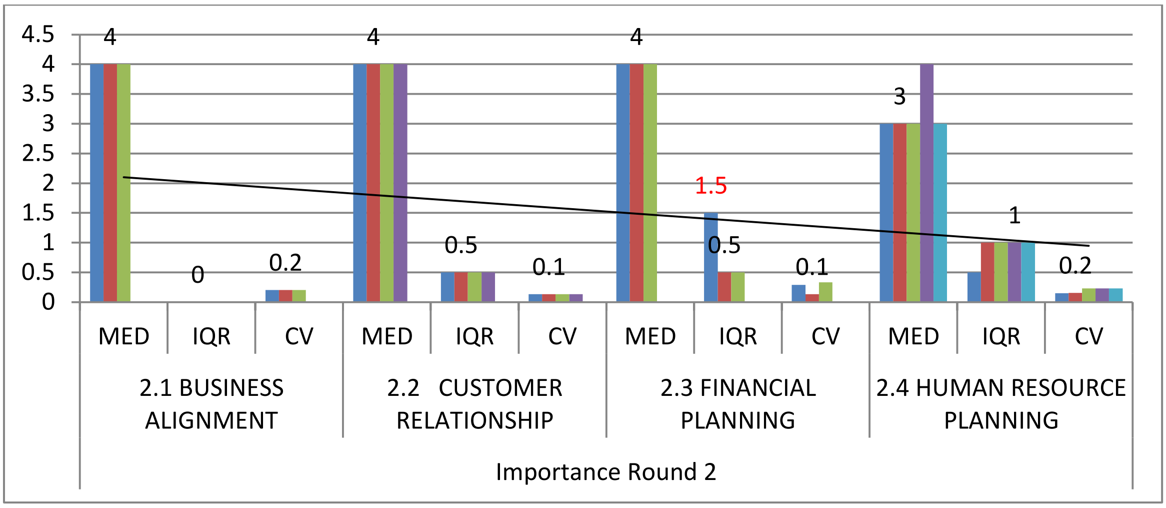


Figure 5-10: Second round ORG results for IMPORTANCE

The realistic explanation of panellists' positive attitude towards *2.2 Customer Relationship* is their strong belief that SSE's attentiveness to customers' needs will contribute largely to the success of the firm and is seen to provide them with a competitive advantage. This view is evident from the following comments provided in the first round:

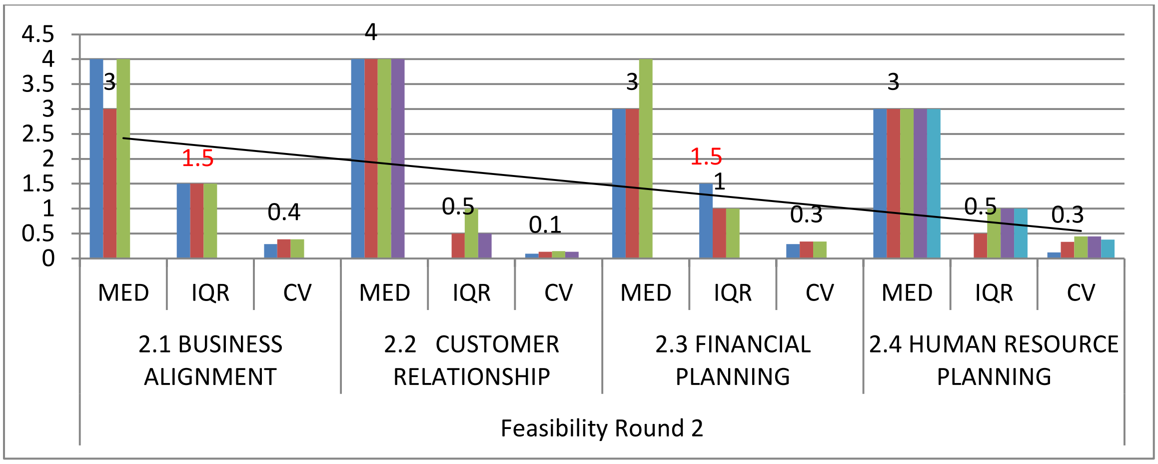


Figure 5-11: Second round ORG results for FEASIBILITY

- *Essential to any business, probably should have priority 1 (#14).*
- *Customers are important for the success of the business. Organisations who are unable to perform Customer Relationship process would fail (#4).*
- *It is Important to maintain client relationship to be able to anticipate and meet current and future needs proactively. Understanding clients' needs and direction can be a competitive advantage (#3).*
- *Equally important is agreeing on the services and service level targets to manage client expectations and potential resource constraints which many SSEs face (#12).*

This view is shared by Alexandre et al. (2006) who, reporting on the characteristics shared among most SSEs, found that customer supplier practices that are legally and contractually imposed are of high quality. This view is consistent with the empirical findings of A Cater-Steel (2004a), who also found that practices related to customer involvement were widely adopted by the 22 firms studied. Likewise, this confirms the findings of Fening et al. (2008) reported above.

Furthermore, the second round results revealed a drop in panellists' ratings of feasibility for *2.2 Human Resource Planning* from very highly (4) to highly (3). Panellists expressed this view in their provided comments emphasising that SSEs cannot afford to send staff for training due to human resources constraints. Alexandre et al. (2006) report on the characteristics of SSEs stresses that training and professional development are very limited due to budget constraints, are only performed to fix technical or performance gaps, and in most cases are non-existent. On the other hand and as per the literature review in chapter 2, empirical studies found selection and development of employees contributes significantly to an organisation's success (Lussier & Pfeifer, 2000).

- *Talent management and professional development are more formally pursued in big organizations and more informal in smaller SSEs. Sometimes, the budget does not allocate for training. My experience in SSE has been that I learned on the job because there was either no budget or no 'convenient time' due to resource constraints (#12).*

- *May not be feasible due to time and budget constraints (#9).*
- *Whilst it is important, SSEs have to balance staffing against training. They should do it as the need arises and have staff availability (#13).*
- *It can be difficult for SSEs to plan, deliver and maintain staff training if they do not have the backup resources to complete the work whilst employees are away for training (#3).*
- *Staff high turnover [very likely in SSEs], tight budget and schedule constraints make it infeasible (#10).*

5.9.2.2 *Dynamic Capabilities*

The DC literature discussed in chapter 2 confirms that all types of organisations small or large are in need of effective capabilities that will enable them to identify new business opportunities and changing customer needs as they happen, then react to these challenges fast and appropriately before the strategic significance is lost (Eisenhardt, 2000; Kivelä, 2007; Teece, Pisano, et al., 1997; Zollo & Winter, 1999). Panellists' comments pointed this out.

The literature review discussed factors that influence the growth and success of SSEs such as strategic planning. There is evidence that lack of strategic planning is linked to SSEs' poor performance (Fening et al., 2008).

Therefore, the DCs identified for ORG capability are firm level capabilities selected to help the enterprise achieve the best strategic fit by investigating changes in technological and regulatory domains and identifying new business opportunities and threats. This information will be the input for defining the organisation's strategic plan. Moreover, ORG capability supports the TSD capability by achieving business IT alignment leading to the development of successful products and services. Therefore, sense, seize and transform processes are adopted in most of the sub processes of ORG capability.

Furthermore, the second group of DCs is focused on the customer/supplier relationship and aims at helping the business grow through sensing opportunities for new customers, negotiating contractual terms and the signing of contracts and through improving the customer's experience by collecting and analysing feedback and taking actions to improve the service.

Finally, DCs have been identified that focus on the human resource aspect in terms of selection and professional development of employees to achieve organisation benefits as discussed in the section above.

5.9.2.3 *Classification of Sub Processes*

Panellists identified all the ORG sub processes as basic practices, core to SSEs. Results show high consensus (N= 71% to 100%). This result is indicative of how panellists' valued the adoption of ORG processes by SSEs.

5.9.2.4 *Final Changes to the model*

Other than the removal of the knowledge management process, only a few minor refinements were implemented in the second version of the model. Table 5-8 provides a presentation of both versions.

Table 5-8: Transformation of the ORG Capability from Version 1 to Version 2

	Dynamic Capabilities	Classification
2.1 BUSINESS ALIGNMENT		
Round 1		
2.1.1 Identify vision and goals, develop business objectives	Sense	
2.1.2 Identify business outcomes to achieve strategic objectives	Seize	
2.1.3 Achieve strategic IT plans.	Transform	
Round 2		
1.1 Develop strategic business objectives	Sense	B
2.1.2 Identify business outcomes to achieve strategic objectives	Seize	B
2.1.3 Implement strategic plan	Transform	B
2.2 CUSTOMER RELATIONSHIP		
Round 1		
2.2.1 Maintain customer relationships	Sense	
2.2.2 Sign up contracts	Size, Transform	

2.2.3 Customer satisfaction survey	Sense	
2.2.4 Analyse customers' surveys and develop action plans	Seize, Transform	

Round 2

2.2.1 Maintain customer relationships	Sense	B
2.2.2 Sign up contracts	Seize, Transform	B
2.2.3 Carry out customer satisfaction survey		B
2.2.4 Analyse surveys and develop action plans.	Sense, Seize	B

2.3 FINANCIAL PLANNING

Round 1

2.3.1 Financial planning	
2.3.2 Develop IT budget	
2.3.3 Financial analysis and reporting	
2.3.4 Invoicing	

Round 2

2.3.1 Develop IT budget		B
2.3.2 Perform financial analysis and reporting	Sense	B
2.3.3 Establish invoicing system		B

2.4 HUMAN RESOURCE PLANNING

Round 1

2.4.1 Identify needed skills and competencies.	Sense
2.4.2 Recruit qualified staff	Seize, Transform
2.4.3 Evaluate and develop staff skills	
2.4.4 Plan training needs.	Sense
2.4.5 Deliver and maintain staff training	Transform

Round 2

2.4.1 Identify needed skills and competencies	Sense	B
2.4.2 Recruit qualified staff	Seize, Transform	B
2.4.3 Evaluate and develop staff skills		B
2.4.4 Plan training needs	Seize	B
2.4.5 Deliver and maintain staff training	Transform	B

2.5 KNOWLEDGE MANAGEMENT

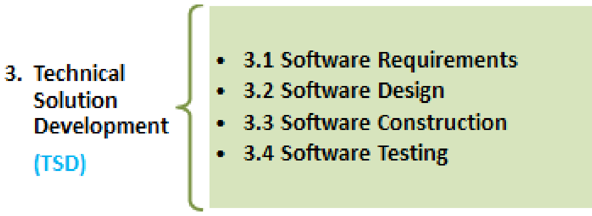
Round 1

2.5.1 Develop KM infrastructure	
2.5.2 Establish a network of experts	Sense
2.5.3 Capture and Disseminate knowledge	Seize, Transform
2.5.4 Maintain knowledge system	Sense, Seize

Round 2

Knowledge management process was dropped in the second round

5.9.3 Technical Engineering



5.9.3.1 *Analysis of Results*

This capability covers the process areas that apply to the product development lifecycle and it consists of four processes *3.1 Software Requirements*, *3.2 Software Design*, *3.3 Software Construction* and *3.4 Software Testing*.

Since this capability is directly linked to the development and implementation of the software product, there was high agreement among panellists for importance on almost all the sub processes in the first round (IQR=0 to 0.1), and all were rated very highly important (N >= 86%, MED = 4). Additionally, there was a good degree of stability in results (0 to 0.2) for the majority of sub processes. Results are presented in Figure 5-12; the full set of results is provided in Appendix D.

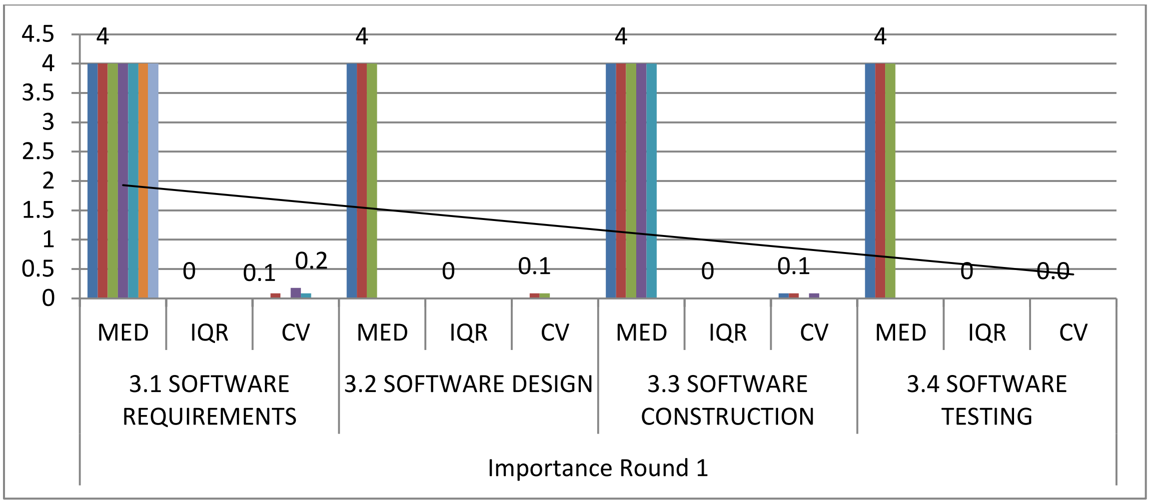


Figure 5-12: First round TSD results for IMPORTANCE

The majority of qualitative comments provided by panellists on the TSD sub processes were recommendations for changes such as combining sub processes or changing the process name.

- *Based on your description of this process, you may want to rename it to 'Develop technical specifications' as opposed to 'Analyse requirements' (#12).*
- *Given the number of tests that could be conducted and with the fact that each software development initiative is slightly different, it may be useful to have a process called 'Define Test Strategy' which, outlines the test environment (#9).*

These very positive results for TSD should come as no surprise and are in agreement with the findings of the literature review of this study. Research studies reported that SSEs have higher competence in technical capabilities than others particularly those related to testing and quality assurance processes (Alexandre et al., 2006; Habra, Alexandre, Desharnais, Laporte, & Renault, 2008). Moreover, empirical evidence shows that processes related to technical capabilities are widely adopted by SSEs compared to management and support (A Cater-Steel, 2004a, p. 243). On the other hand, studies also reported that in SSEs TSD processes still rank low compared to large software enterprises (LSE) (Alexandre et al., 2006). This flags the need to increase the maturity level of their TSD processes.

Similar positive high ratings were expressed for importance in the second round with a good degree of stability (CV = 0.2). However, three sub processes, 3.3.1 Develop coding standards, 3.3.2 Obtain complete test cases and 3.4.2 Perform regression testing, showed a slight lack of consensus (IQR = 1.5). Results are shown in Figure 5-13 and the full quantitative set is provided in Appendix D.

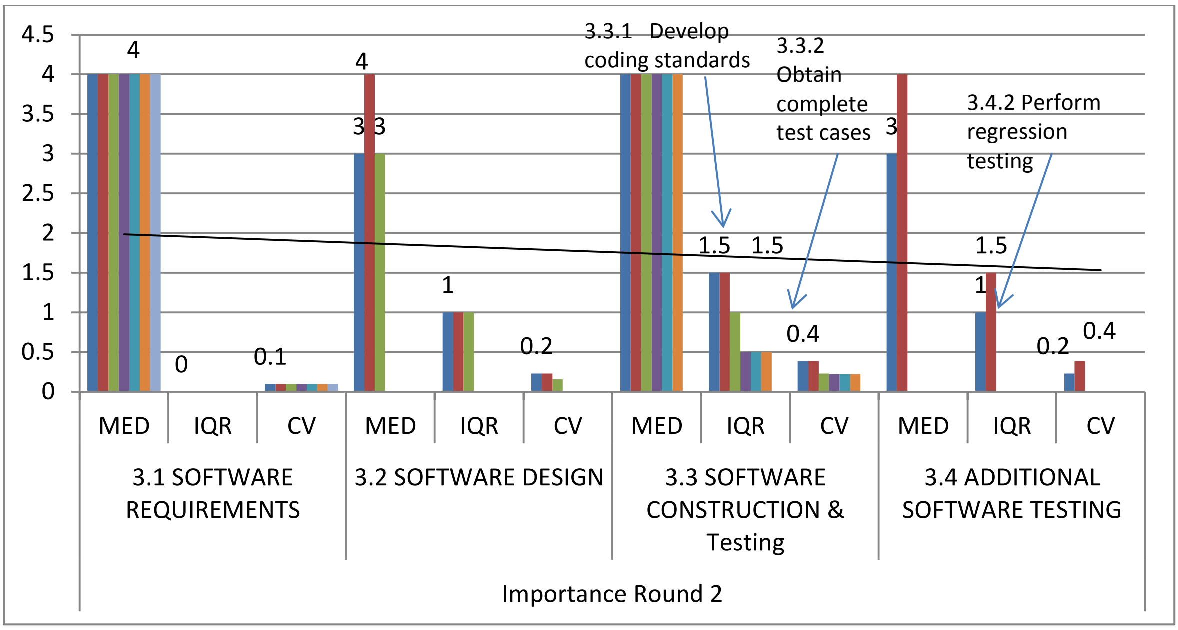


Figure 5-13: Second round TSD results for IMPORTANCE

The slight disagreement in the second round on the three sub processes can be understood from reading the comments provided by panellists in the first round shown below:

- *[Develop coding standards] is important for code comprehensibility but it is hard to bring all the developers to follow standards as everyone has individual coding style (#4).*
- *For small changes, a full suite of regression testing is not always done because it may push project timelines unless regression testing is automated (#9).*
- *Most of the time test cases are not updated due to time constraints and an ad-hoc approach is adopted (#3).*

As shown in Figure 5-14 the analysis of feasibility results for the first round shows consensus (IQR=1), although slightly lower than importance, and also shows a high percentage of very highly feasible ratings (N >= 67%). Moreover, there is a good degree of stability in results (CV= 0.3). These findings assert that panellists considered this capability highly feasible for SSEs.

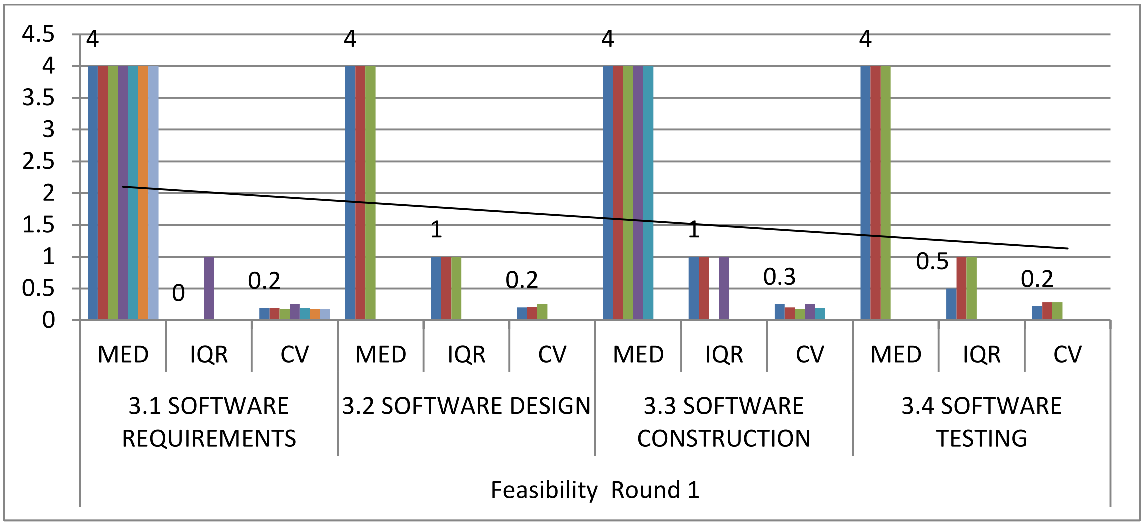


Figure 5-14: First round TSD results for FEASIBILITY

As can be seen from Figure 5-15, the second round feasibility results maintained consensus for all the sub processes with a good degree of stability in ratings (CV = 0.3). However, the overall ratings of some sub processes dropped from very highly feasible to very feasible (MED =3). *3.3.2 Obtain complete test cases* sub process show a slight lack of consensus (IQR=1.5). From the analysis of qualitative comments the fact that 3.3.2 lacked consensus is because some panellists support the application of agile software development methodology, which does not require extensive documentation, including developing test cases.

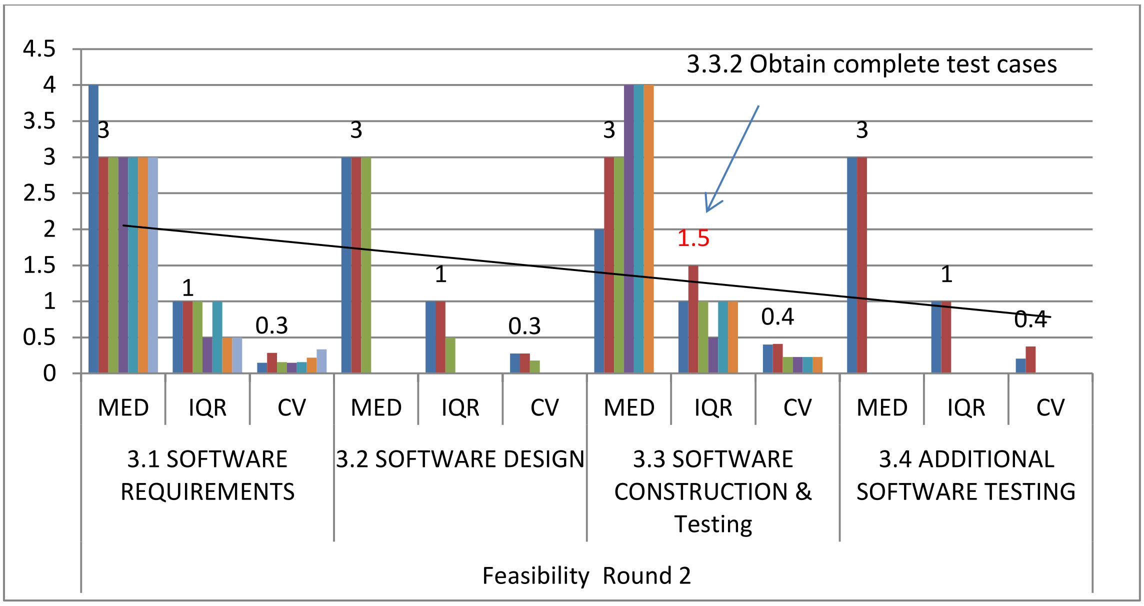


Figure 5-15: Second round TSD results for FEASIBILITY

Reflecting on the overall capability ratings in the second round, we can see a drop in TSD ratings from very highly (4) to highly (3) for both importance and feasibility. Meanwhile, a noticeable shift in panellists' opinion was obvious for ORG and SPD&I, as there was an increase in the high ratings for some of the processes in regards to importance and feasibility. This observation implies an increase in awareness among panellists towards ORG and PD&I capabilities, which made them reassess their ratings for some TSD sub processes, and in order to make it feasible for SSEs to allocate resources to other more important sub processes from the other capabilities, the ratings for some TSD processes dropped.

5.9.3.2 *Dynamic Capabilities*

There are no DCs identified for TSD; all the processes and sub processes are considered operational processes that are performed to develop the software product. The difference between operational and dynamic capabilities are discussed in chapter 2, section 2.5.5 on page 45.

5.9.3.3 *Classification of Sub Processes*

As demonstrated in Table 5-9 and the results in Appendix D, panellists expressed high consensus on the classification of all TSD sub processes as basic practices.

5.9.3.4 *Final Changes to the model*

The TSD has undergone changes at the end of the first round based on panellists' recommendations and addressed concerns. A detailed presentation of the changes as well as justifications can be found in the second round feedback sheet in Appendix C. This was distributed to panellists in the second round. Following are the first and second round versions of the model.

Table 5-9: Transformation of the TSD Capability from Version 1 to Version 2

3.1 SOFTWARE REQUIREMENTS		
Round 1	**Dynamic Capabilities**	**Classifications**
3.1.1 Obtain software scope		
3.1.2 Elicit requirements		
3.1.3 Analyse requirements		
3.1.4 Trace requirements		
3.1.5 Validate and prioritise requirements		
3.1.6 Develop requirements baseline		
3.1.7 Develop test cases		
Round 2		
3.1.1 Obtain software scope		B
3.1.2 Elicit requirements		B
3.1.3 Develop requirements specifications		B
3.1.4 Validate and prioritise requirements		B
3.1.5 Establish requirements baseline		B
3.1.6 Develop testing strategy		B
3.1.7 Develop test cases		B

3.2 SOFTWARE DESIGN		
Round 1		
3.2.1 Develop software architectural		
3.2.2 Verify design		
3.2.3 Update test cases		
Round 2		
3.2.1 Develop software architectural		B
3.2.2 Verify design		B
3.2.3 Update test cases		B

3.3

Round 1

3.3 SOFTWARE CONSTRUCTION	
3.3.1 Develop coding standards	
3.3.2 Obtain complete test cases	
3.3.3 Code software product	
3.3.4 Verify individual software units	
3.3.5 Verify the integrated units	

Round 2

3.3 SOFTWARE CONSTRUCTION & Testing		
3.3.1 Develop coding standards		B
3.3.2 Obtain complete test cases		B
3.3.3 Code software product		B
3.3.4 Verify individual software units		B
3.3.5 Verify the integrated units		B
3.3.6 Validate the System		B

3.4

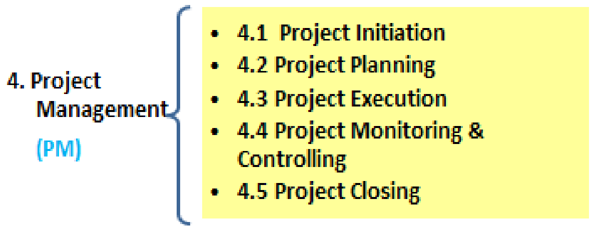
Round 1

3.4 SOFTWARE TESTING	
3.4.1 Apply system performance testing	
3.4.2 Validate the system	
3.4.3 Perform regression testing	

Round 2

3.4 ADDITIONAL SOFTWARE TESTING		
3.4.1 Apply testing		B
3.4.2 Perform regression testing		B

5.9.4 Project Management



5.9.4.1 *Analysis of Results*

The project management capability is structured using the five process groups proposed by the Project Management Body of Knowledge (PMBOK) as listed in the diagram above. The full list of quantitative results is provided in Appendix D.

As can be seen from Figure 5-16, and the first round results for importance listed in Appendix D, there is an apparent collective consensus among panellists with very highly important rating for most sub processes (IQR=0-1, N=57% to 89%,MED = 4). A good degree of stability in results is shown for all the sub processes.

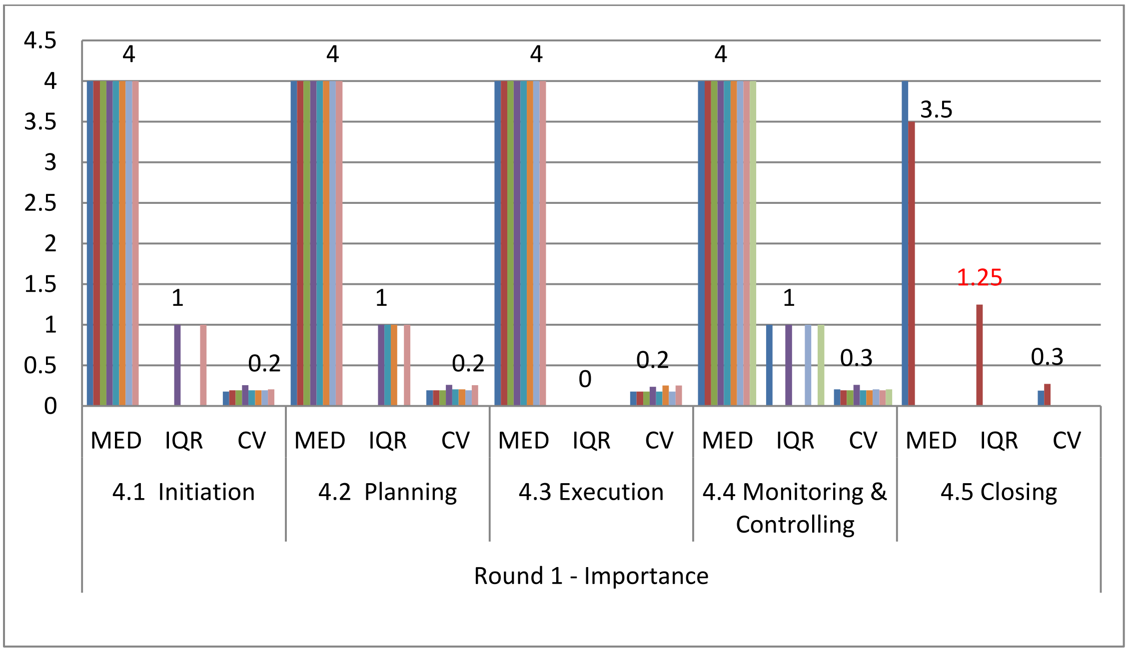


Figure 5-16: First round PM results for IMPORTANCE

The qualitative responses by panellists' in the first round conveyed a collective view (N=44%) that the proposed PM capability is complex and must be simplified using more practical sub processes. Therefore, PM has undergone major changes resulting in sub processes being more aligned with PMBOK processes. Following are excerpts of panellists' comments:

- *[PM] capability is very important for SSEs, the processes are too detailed and should be collapsed. It is important not to lose sight of the essence and the value by adding too many sub processes (#13).*
- *Reasonably good; however, it is critical to know that SSEs are unlike large organisations, and cannot sustain a model with too many processes. The current structure can be time consuming and infeasible for them (#11).*

Panellists provided recommendations on how to restructure the model:

- *All these processes are important but could be simplified into practices that are more practical. For example:*

  4.1 Project Initiation

  4.1.1 Develop Business Case / Feasibility Study

  4.1.2 Develop Project Proposal (Includes high-level project goal, scope, schedule, cost, risk analysis, project resource requirements based on skill set (#12)

- *Quality activities may be embedded as part of requirements management and testing on small to mid-size initiatives (#4).*
- *Project monitoring and controlling are shadow activities that occur throughout the project life cycle. As such, the data that feeds into these activities can be captured in the initiation, execution and close phases. The activities could be collapsed into one, simply called 'Project Monitoring and Reporting' with the standard that it focuses on reporting change, scope, schedule, cost, HR, risk and quality (#12).*

As for feasibility results, it can be seen from Figure 5-17 below that the first round results show disagreement on most sub processes (IQR = 2), despite the fact that a number of panellists still consider the processes highly feasible for SSEs (MED = 4 for most sub processes). The full set of results is presented in Appendix D.

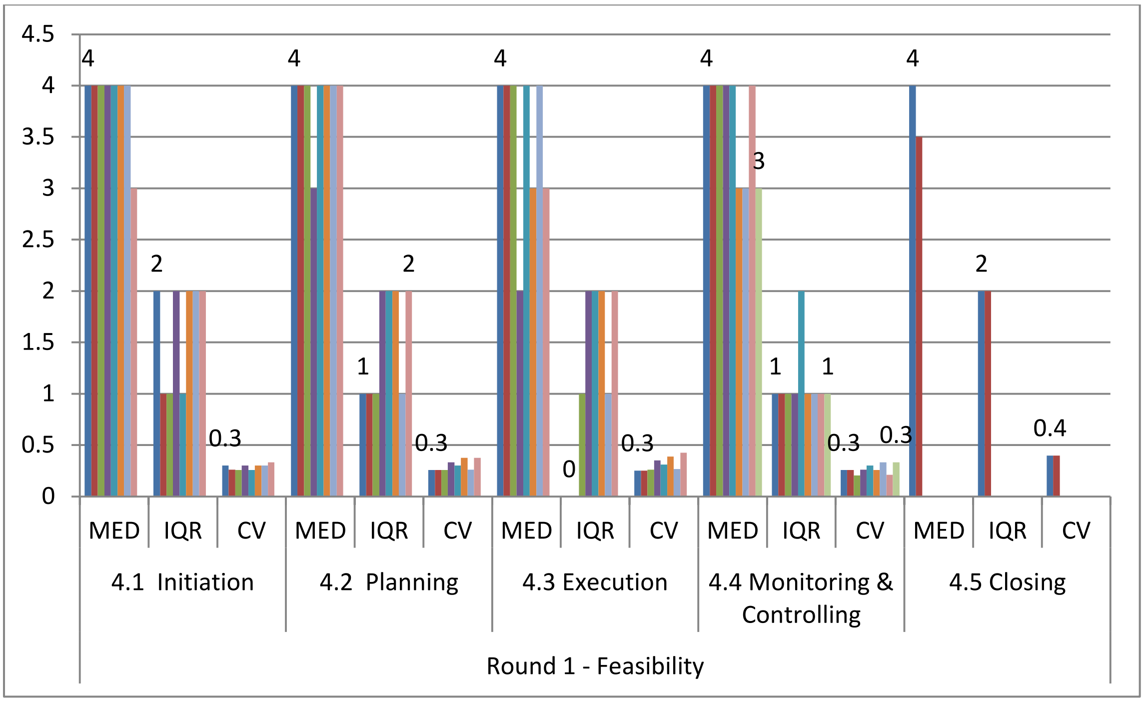


Figure 5-17: First round PM results for FEASIBILITY

This disagreement can be attributed to the perceived complex structure of the proposed capability, rather than the infeasibility of PM capability in general. In fact, the analysis of first round qualitative data supports this conclusion. Below are excerpts from panellists' comments provided in the first round.

- *Again, [PM capability] is very important for a successful business but for a small business it often becomes hard to achieve completely and so a trade-off analysis is done (#9).*
- *[PM capability] is comprehensive (#3).*

In the second round, as can be seen from Figure 5-18, it is clear that a consensus has been reached on most of the PM sub processes apart from three processes that show slight disagreement (IQR=1.5). This shows that almost all panellists perceived PM capability as very highly important for SSEs (N=57% to 71%, MED = 4). In addition, a good degree of stability can be seen for all the sub processes (CV= 0.2 or 0.3), which confirms that panellists collectively accepted the proposed changes in the second round, which dismisses the need for another round. Refer to Appendix D for full result set.

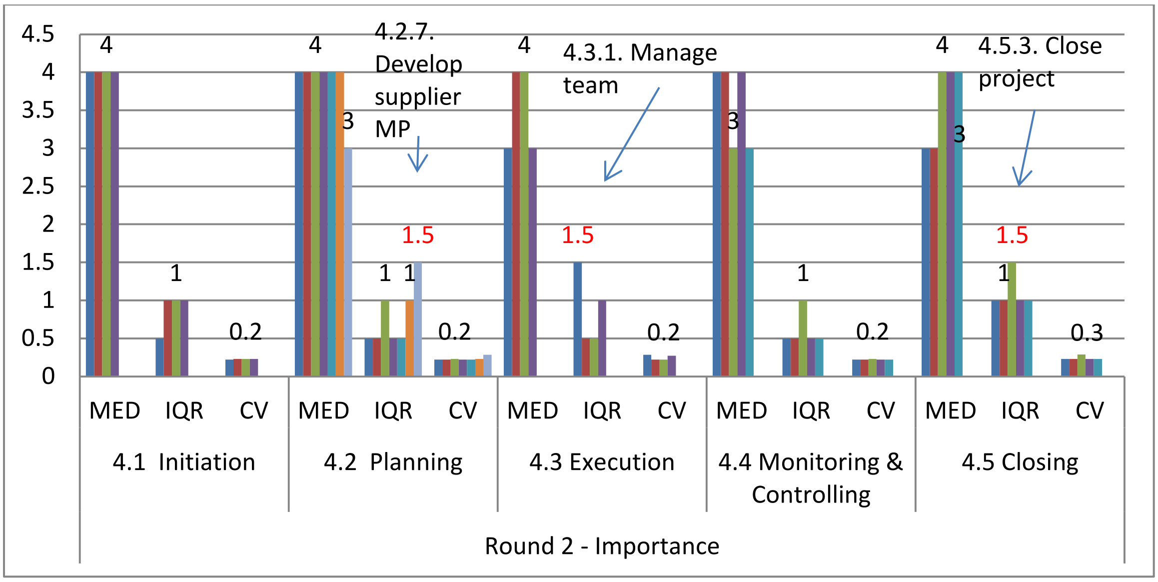


Figure 5-18: Second round PM results for IMPORTANCE

The second round results presented in Figure 5-18, 4.2.7 Develop Supplier Management Plan and *4.3.1 Manage team* show slight disagreement and are considered just important (rating =2) by a few panellists. This may be because it was assumed that communication between stakeholders and team members could happen sufficiently well in SSEs due to the small size of teams. Hence, explicit work procedures seem not very necessary. As for *4.5.3 Project closure*, some panellists expressed their view that due to the extensive workload documentation of project performance is unlikely to happen. Following are comments provided in the first round that support this view:

- *Smaller teams tend to collaborate and communicate more informally and frequently. Formal communication could be simplified through weekly team status and monthly project reporting to the sponsor or steering committee (#3).*
- *[For initiating supplier management], most SSEs have an ad-hoc way of managing their suppliers and it is sometimes rather an informal engagement (#9).*
- *[Update project documentation is] infeasible to keep it up-to-date for SSEs because the same person may be responsible for other tasks too (#14).*
- *[Risk Management] takes extra resources so is often partially ignored (#4).*

As for feasibility results in the second round, we can see from Figure 5-19 and the results in Appendix D, that a consensus has been reached for all the sub processes except for *4.3.2. Conduct progress review meeting* and *4.1.3. Develop stakeholder register and management strategy,* which both show slight disagreement (IQR =1.5). Therefore, feasibility findings are also indicative of panellists' collective approval of the changes proposed for PM in the second round.

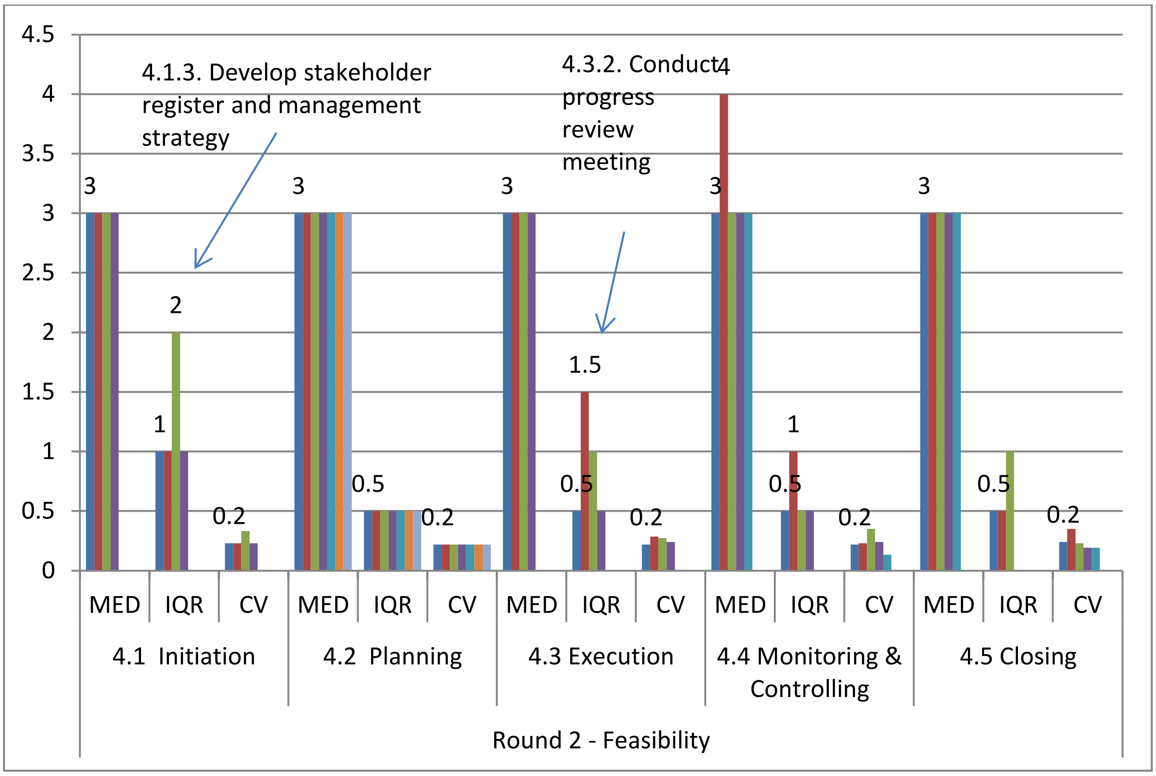


Figure 5-19: Second round PM results for FEASIBILITY

In light of what was discussed above, it is worth shedding light on the findings of Lester et al. (2010) study of ten small to medium software enterprises (MSE). Their findings concerning PM capability revealed that all the firms they appraised performed project planning at a basic level, whereas SSEs lacked processes that involved documentation of lessons learned, recording of historical data and re-planning due to deviations. Similarly, project monitoring and controlling processes were almost equally adopted in SSEs and MSEs, but SSEs were seen to perform markedly lower in monitoring project risks and conducting milestone reviews. Hence, the findings of Lester et al. (2010) show that SSEs are better at monitoring project progress and reacting to arising problems than proactive sensing of issues early in the project lifecycle, an attribute common among organisations with low maturity level. This explains the disagreement by panellists on processes that involved proactive practices such as stakeholder management strategy, conducting progress review meetings along with the collection of historical data as part of the project closure process.

5.9.4.2 *Classification of Processes*

The analysis of quantitative results revealed high consensus (IQR=0) on the classification of PM sub processes as basic processes. This explains the high perception held by the majority of panel members in regards to this capability.

5.9.4.3 *Dynamic Capability*

The purpose of dynamic capabilities for PM is to help SSEs in identifying and responding to project level opportunities and threats. SSEs are more likely to implement project level DCs than organisation level DCs, to help them respond to customer requirements of meeting deadlines and avoiding project related risks.

The model included DCs for all PM processes. The first round information sheet provided in Appendix C includes the first version of PM processes. The majority of panellists (N= 88%) responded with Yes to questions about their agreement with the proposed PM DCs. No comments were given; this is interpreted as consensus on the usefulness of the identified DCs for PM. The general comments on DCs support this conclusion; following are excerpts:

- *DCs are needed by SSEs as it will enable them to respond fast to rapid changes in technology, business opportunities and customer needs as they happen (#10).*
- *I agree that the dynamic capabilities are generally relevant in a software development process but not necessarily crucial to enable success for SSE (#12).*

5.9.4.4 *Final Changes to the model*

As discussed in the previous section, PM capability has entirely changed as a response to panellists' recommendation for a simpler and more practical set of processes. Table 5-10 provides a presentation of the differences between the first and second round versions of the model.

Table 5-10: Transformation of the PM Capability from Version 1 to Version 2

4.1 PROJECT INITIATION		
Round 1	**Dynamic Capabilities**	**Classifications**
4.1.1. Perform feasibility analysis	Sense, Seize	
4.1.2. Initiate scope		
4.1.3. Initiate schedule		
4.1.4. Initiate budget		
4.1.5. Initiate communication management		
4.1.6. Initiate team		
4.1.7. Initiate risk analysis	Sense	
4.1.8.Initiate supplier management	Sense, Seize, Transform	
Round 2		
4.1.1. Develop business case	Sense, Seize	B
4.1.2. Develop project charter	Transform	B
4.1.3. Develop stakeholder register and management strategy		B
4.1.4. Develop supplier proposals	Sense, Seize, Transform	B

4.2 PROJECT PLANNING		
Round 1		
4.2.1 Plan scope		
4.2.2 Plan schedule		
4.2.3 Plan cost		
4.2.4 Plan quality		
4.2.5 Plan communication		
4.2.6 Plan human resources		
4.2.7 Plan Risks	Seize	
4.2.8 Plan supplier management	Seize	
Round 2		
4.2.1 Develop scope		B
4.2.2 Develop schedule		B

4.2.3. Develop responsibility matrix		B
4.2.4. Develop risk management plan	Seize	B
4.2.5. Develop budget		B
4.2.6. Develop project communication plan		B
4.2.7. Develop supplier management plan	Seize	B

4.3 PROJECT EXECUTION

Round 1

4.3.1 Execute scope	
4.3.2 Execute schedule	
4.3.3 Execute cost	
4.3.4 Execute quality management plan	Sense, Seize, Transform
4.3.5 Perform communication management	Sense, Seize, Transform
4.3.6 Execute human resource management	
4.3.7 Execute risk management plan	Sense, Seize, Transform
4.3.8 Perform supplier management	Sense, Seize, Transform

Round 2

4.3.1. Manage team		B
4.3.2. Conduct progress review meeting	Sense, Seize	B
4.3.3. Update project tracking sheets		B
4.3.4. Perform Supplier management	Sense, Seize	B

4.4 PROJECT MONITORING & CONTROLLING

Round 1	DC Sense, Seize, Transform
4.4.1 Integrated change control	
4.4.2 Monitor and control scope	
4.4.3 Monitor and control schedule and cost	
4.4.4 Monitor and control cost	
4.4.5 Monitor and control quality	
4.4.6 Monitor and control communication	
4.4.7 Monitor and control human resources	
4.4.8 Monitor and control risks	
4.4.9 Monitor and control supplier	

Round 2	DC Sense, Seize, Transform	
4.4.1. Assess scope changes		B
4.4.2. Monitor and control schedule and cost		B
4.4.3. Monitor and control Team		B
4.4.4. Monitor and control risks		B
4.4.5. Monitor and control supplier		B

4.5 PROJECT CLOSING

Round 1

4.5.1 Close supplier contracts	
4.5.2 Update project documentation	Sense, Seize

Round 2

4.5.1. Run project closure meeting(s)		B
4.5.2. Close supplier contracts		B
4.5.3. Close project	Sense, Seize	B
4.5.4. Plan project release		B
4.5.5. Deploy release		B

5.9.5 Support

5. Support
(SUP)

- **5.1 Service Level Management**
- **5.2 Change Management**
- **5.3 Release and deployment**
- **5.4 Configuration Management**
- **5.5 Problem Management**

5.9.5.1 *Analysis of Results*

The intended benefit of the SUP capability is to strengthen SSE’s accountability for post-product maintenance services to ensure its maintainability and availability to end users. The Literature stresses that software support is a neglected area and it is handled by software development organisations as part of the TSD capability being the final phase of the software development lifecycle. Empirical literature stresses the need to recognise the unique nature of software support and the need for separate effective processes to handle this capability efficiently (Rashid et al., 2009, p. 45). Therefore, five processes were included in the model.

Following is a discussion of the results, the full set of quantitative results is provided in Appendix D.

In this round the majority of panel members expressed agreement for all SUP sub processes (IQR = 0.5 or 1) and rated importance very highly (MED=4). Moreover, results showed acceptable level of stability for results (CV=0.3). These results are presented in Figure 5-20.

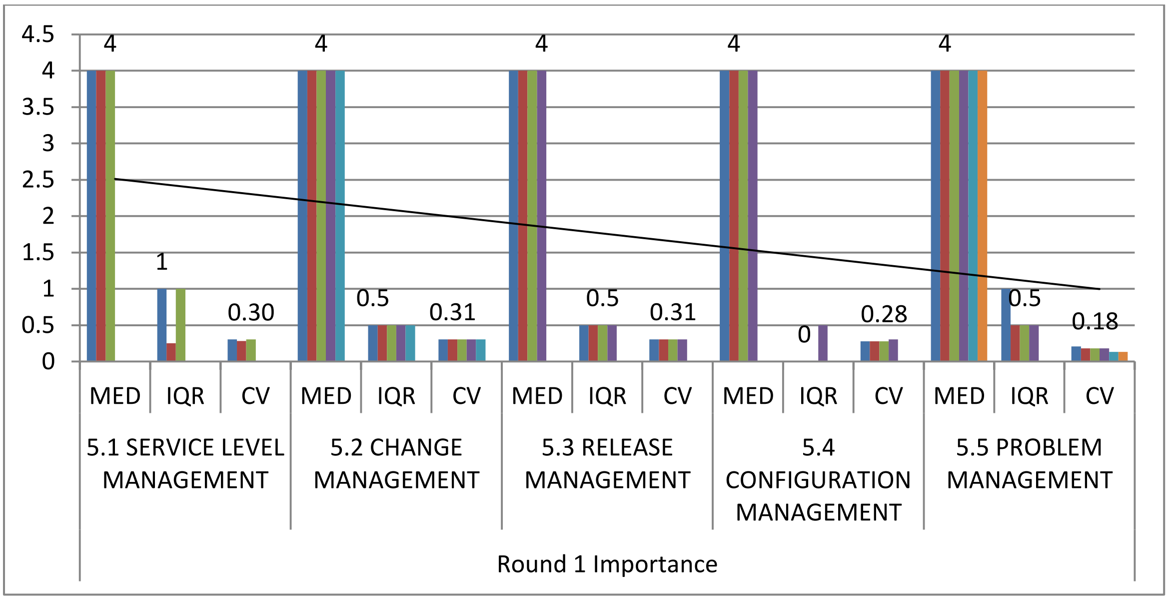


Figure 5-20: First round SUP results for Importance

Furthermore, the analysis of the comments shows that panellists consider this capability critical for sustaining the business and for providing on-going product value for customers. This is demonstrated in the following comments:

- *I think this capability is important to maintain the ongoing relevance of the software product and to ensure it continues to meet the business needs (#11)*
- *[SUP capability] is important to sustain the products and business and to deliver high quality maintenance (#9).*

As for feasibility, the analysis of quantitative data in the first round shows disagreement on the voted ratings for the majority of sub processes as presented in Figure 5-21. Moreover, feasibility results also showed an unacceptable level of stability for a few sub processes (CV>=0.6). The two processes that received high disagreement are *5.3 Change Management* and *5.4 Configuration Management (IQR=2).* All 5.4 *Configuration Management* sub processes showed disagreement as panellists believed that configuration management should be given lowest priority by SSEs when compared to other competing business matters.

Following are the comments provided by panellists to justify their ratings for CM:

- *Full identification [of configuration items] is not possible (#12).*
- *Small organisations do not have the resources to invest in a CMS (#4).*
- *Time at hand is spent on new projects/issues rather than configuration audits, which become a lower priority (#14).*

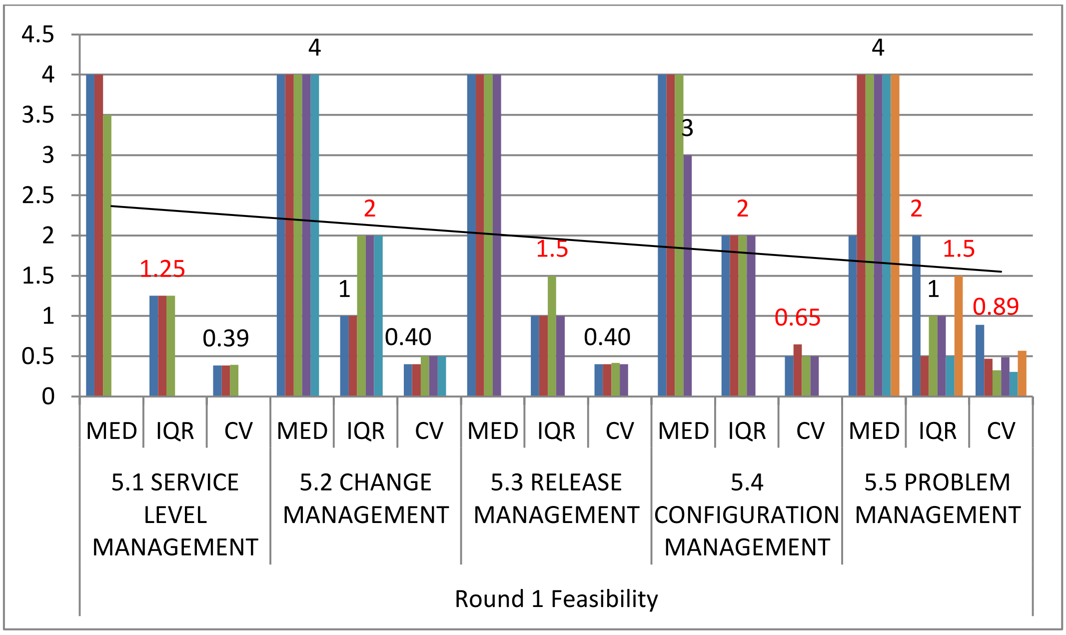


Figure 5-21: First round SUP results for Feasibility

Additionally, the results for *5.2 Change Management* showed disagreement for three processes 5.2.3, 5.2.4, 5.2.5 as shown by Figure 5-21. However, the review and evaluation of change requests (5.2.1 and 5.2.2), which are the most important sub processes, have received a very high feasibility rating.

Furthermore, there appears to be slight disagreement (IQR=1.25) for all the sub processes underlying *5.1 Service Level Management*, although this is a core process for this capability. However, as demonstrated by panellists' qualitative responses presented below, they have underscored the importance of *5.1 Service Level Management* process, but also stressed the informal nature of conducting this process by SSEs due to the unavailability of a dedicated experienced team to handle it. Research reports that inexperienced developers normally handle software support and maintenance activities, and often they are not assigned to dedicated teams. (Rashid et al., 2009, p. 48). SWEBOK guide considers service level agreement as the most important support process (Abran, Boruque, & Trip, 2004).

Following are panellists' comments concerning process 5.1:

- *This is critical for SSEs as many of them will do "extra" services to keep their customers happy and some customers take advantage of that. Agreements can help get the service back to the agreed scope (#7)*
- *These are all important but typically not formally conducted in SSEs as the project team members usually become the support team once the software cuts over to production. It is however important to ensure service level targets and client expectations are agreed upon (#12).*

As can be seen in Figure 5-22, in the second round, results interestingly revealed agreement (IQR<=1) on the importance of SUP processes with ratings very highly important (MED =4). The only sub process that showed disagreement is *5.4.5 Post Implementation Review and Closure,* as it was rated low by few a panel members (N=25%). Panellists did not provide comments to justify their ratings of this sub process. However, from our PM findings and discussion in the above section, and reflecting on the findings of Lester et al. (2010), it can be seen that review activities are normally perceived as not entirely feasible for SSEs.

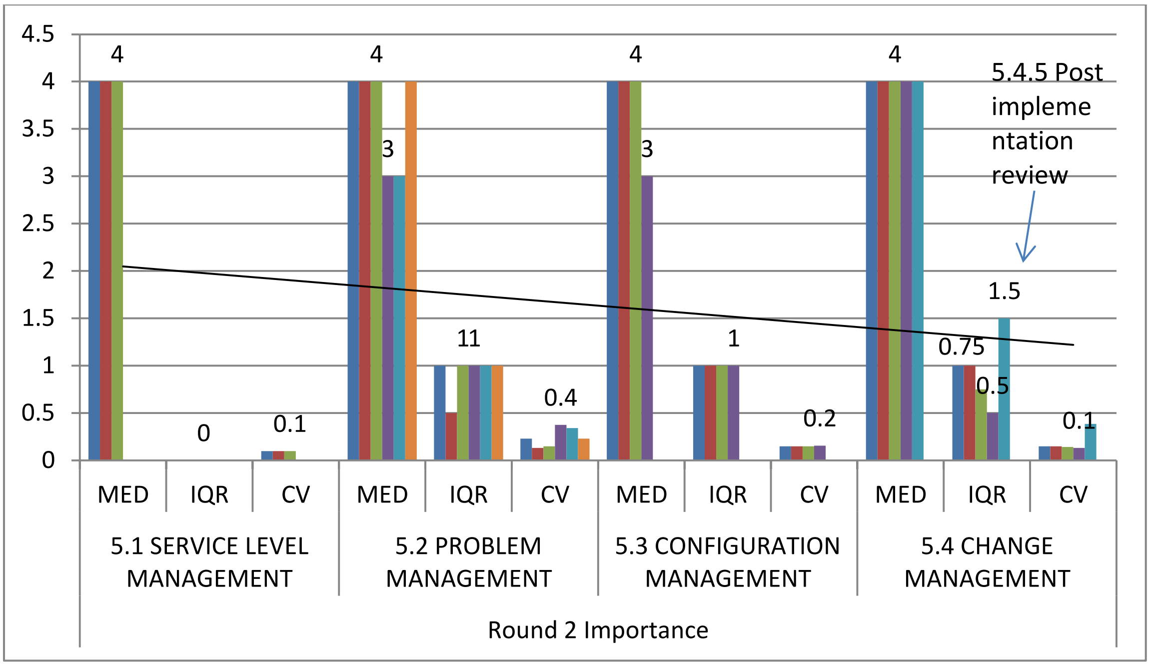


Figure 5-22: Second round SUP results for Importance

As shown in Figure 5-23, feasibility results show high agreement with a slightly lower rating (MED = 3) and a good level of stability for all the sub processes. Results revealed slight disagreement on three sub processes and these are: *5.2.2. Problem categorisation and prioritisation*, *5.2.3 Problem diagnosis and resolution* and *5.3.1 Identify configuration items*. However, none of the three sub processes received a rating below 2. The analysis of qualitative data revealed that panellists generally considered the processes that involve proactive problem resolution, problem closure and review and configuration audits as unfeasible for SSEs. The relevant panellists' comments are presented below:

- *Time at hand is spent on new projects or resolving issues rather than performing configuration audits, which are given, lower priority (#14)*
- *Problems closures do not often happen (#9).*
- *Staff are usually busy creating or fixing other things rather than being involved in proactive problem identification (#13).*

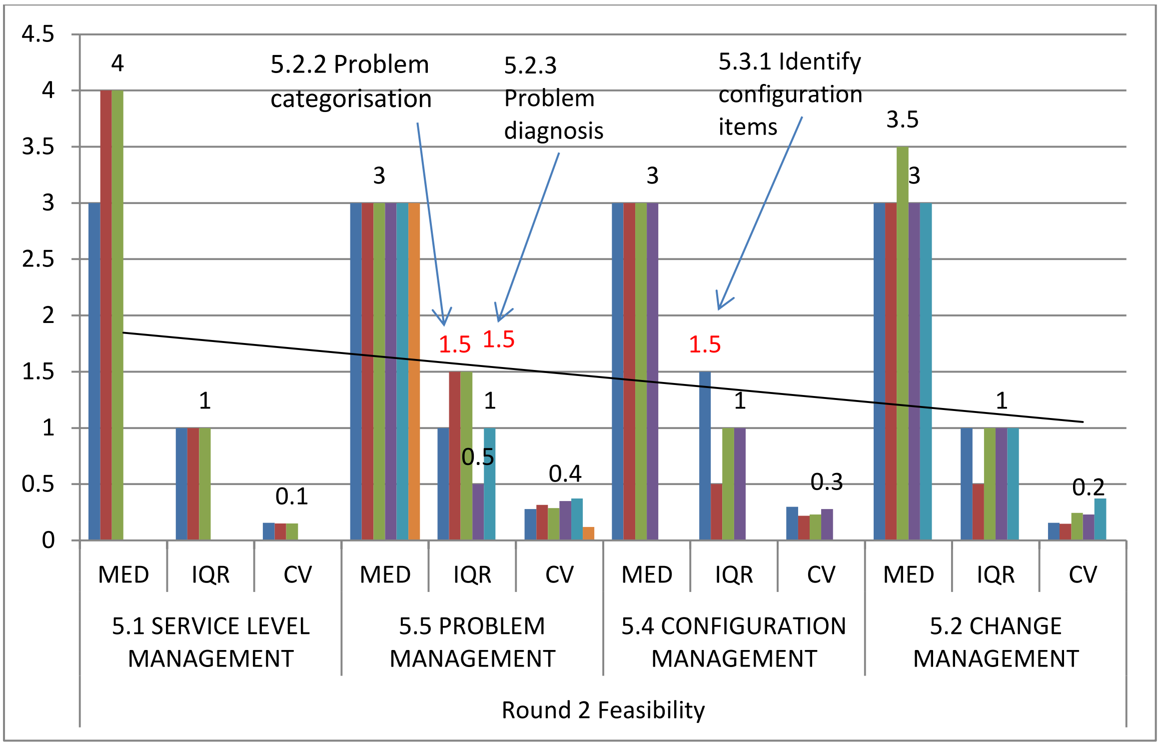


Figure 5-23: Second round SUP results for Feasibility

5.9.5.2 *Classification of Processes*

Results of the second round reveal consensus on the classification of the majority of SUP sub processes as basic practices. Interestingly two sub processes related to configuration management, *5.4.1 Identify configuration items* and *5.3.4 Perform Configuration Audits* were classified as advanced.

5.9.5.3 *Dynamic Capability*

Similar to TSD capability the majority of the SUP sub processes deal with operational business activities. However, unlike TSD, SUP included organisational as well as project level processes. Following are the processes that were regarded as DCs:

5.2.5 Major problem review (Project level DC Sense)	This sub process is responsible for revealing the root causes of problems and performing trend analysis. Hence, it provides inputs to improve the infrastructure, IT services and organisational processes.
5.4.5 Post implementation review and change closure	Post implementation review of major changes especially when breaches to contracts are involved. Root cause analysis to identify mistakes and lessons learned are

(Organisational level DC Sense, Seize)	documented to prevent future issues with similar future changes.
5.3 Configuration Management (DC Enabler)	Depending on the scope of the configuration management database, this process is considered an enabler to DCs when the organisation decides to document the lessons learned from the review of major problems and changes into the configuration management database.

Panellists responded with Yes to all the selected DCs, except one member who was consistently opposing the value of DC processes. Panellists provided no recommendations for removal or additions of DCs, hence version 1 and version 2 of the model retained the same DCs.

5.9.5.4 *Final Changes to the Model for SUP Capability*

The SUP process has undergone minor changes in the first round. Generally, panellists did not provide comments, which is interpreted as acceptance of the suggested processes and sub processes. However, the following comments provided by one panel member gave insights for improvements.

- *The project management capability 'Project Execution' activity must include all the preparatory and transitional activities required to transfer the system to production. These include the engagement of support team, training, define and signing off operational agreement. I disagree with your model where it currently sits in the 'Support Capability'. Support implies production and operations. The project management capability is in essence delivering a solution in production and the 'Project Close' [process] must include the establishment of the solution and the environment as well as the post-implementation review (#12).*

Accordingly, release management sub processes were reduced to two processes (4.5.2 and 4.5.3) and moved to PM project closing process. Table 5-11 is a presentation of both versions of the model, the first round (version 1) and the second round (version 2):

Table 5-11: Transformation of the SUP Capability from Version 1 to Version 2

5.1 SERVICE LEVEL MANAGEMENT		
Round 1	**Dynamic Capabilities**	**Classifications**
5.1.1 Identification of service requirements		
5.1.2 Agreements sign-off		
5.1.3 Service level monitoring and reporting		
Round 2		
5.1 SERVICE LEVEL MANAGEMENT		
5.1.1 Identification of service requirements		B
5.1.2 Agreements sign-off		B
5.1.3 Service level monitoring and reporting		B

5.5 PROBLEM MANAGEMENT		
Round 1		
5.5.1 Proactive problem identification		
5.5.2 Problem categorization and prioritization		
5.5.3 Problem diagnosis and resolution		
5.5.4 Problem closure and evaluation	Seize	
5.5.5 Major problem review	Sense	
5.5.6 Problem management reporting		
Round 2		
5.2		
5.2.1 Proactive problem identification		B
5.2.2 Problem categorisation and prioritisation		B
5.2.3 Problem diagnosis and resolution		B
5.2.4 Problem closure and evaluation	Seize	B
5.2.5. Major problem review (DC- Sense)	Sense	B
5.2.6. Problem management reporting		B

5.4 CONFIGURATION MANAGEMENT	
Round 1	**DC Enabler**
5.4.1 Identify configuration items	
5.4.2 Establish a configuration management system	
5.4.3 Control configuration items	
5.4.4 Perform configuration audits	

5.3 CONFIGURATION MANAGEMENT

Round 2	DC Enabler	
5.3.1 Identify configuration items		B
5.3.2 Establish a configuration management system		B
5.3.3 Control configuration items		B
5.3.4 Perform configuration audits		B

5.2 CHANGE MANAGEMENT

Round 1

5.2.1 Review of request for change	
5.2.2 Assessment of change request	
5.2.3 Change evaluation authorisation	
5.2.4 Change deployment authorisation	
5.2.5 Post implementation review and change closure	Sense, Seize

Round 2

5.4		
5.4.1 Review of request for change		B
5.4.2 Assessment of change request		B
5.4.3 Change Planning		B
5.4.4 Change deployment authorisation		B
5.4.5 Post implementation review and closure	Sense, Seize	B

5.3 RELEASE MANAGEMENT

Round 1

5.3.1 Release planning	
5.3.2 Establish the release infrastructure	
5.3.3 Release documentation	
5.3.4 Deploy the release	

Round 2

In the second version of the model, Release Management is moved to 5.4 Project Closing

5.10 **Summary**

From the quantitative and qualitative analysis it can be concluded that panellists have collectively accepted the second version of the model and have considered the model critical for sustaining organisational success and for enhancing the outcomes of software development projects in SSEs. Meanwhile, a few continued to stress factors related to the inherent characteristics of SSEs that may affect the adoption of some processes in the model. The most emphasised characteristic is the lack of resources and funds, the greater need for flexibility and a flatter structure. The following comment provided by one of the panellists presents this view:

- *Anything related to process is fundamentally critical to enabling and sustaining long-term success. Having said this, I am putting on my executive lens, and putting myself in the shoes of many small software providers/clients I have consulted for, and know that reality is often at odds with idealist views when resources are constrained, as is often the case with small enterprises. My opinions therefore reflect the glaring realities faced by these firms, vs my opinion as to what should be an idealistic or aspirational framework (#4).*

The analysis of panellists rating of the five proposed capabilities revealed that panellists showed support for all five and did not recommend changes or additional capabilities. The quantitative and qualitative data showed that panellists showed highest interest in processes related directly to the product. Hence, TSD and SUP were rated as the most important, followed by PM.

On one hand, the analysis of results for processes revealed that those that received the highest support by panellists are those related directly to customers (customer relationship management and service level management). On the other hand, processes related to strategic planning, software design and review of work (process review, process improvement, project closing, and problem review and closure) are seen as not feasible for adoption by SSEs due to their resource constraints. Moreover, few panellists considered the configuration management process as not feasible for SSEs.

In light of these findings, the qualitative comments provided valuable recommendations for enhancing the adoption of the model. The most emphasised factors included the gradual adoption of the model starting with the processes that are most critical to the organisation, in addition to keeping the process procedures and documentation simple and short.

Finally, panellists expressed their support for the suggested DCs, but only provided supportive comments and there were no recommendations for changes. This can be associated to the fact that due to time constraints stronger emphasis was placed on the evaluation of processes and underlying sub processes and lesser emphasis on DCs; otherwise, it would have led to an unachievable scope.

Chapter 6 - Discussion

6.1 Introduction

In this chapter, the findings of quantitative and qualitative data from the Delphi study are summarised and discussed, and are further validated using prior research findings explored in chapter 2. The chapter answers RQ2, RQ3 and RQ4.

6.2 Discussion

6.2.1 Capabilities Critical for the Model

This section attempts to address the second research question, *What are the specific capabilities critical for the desired model that would benefit SSEs at each phase of the software development lifecycle?* These capabilities, which were identified in chapter 3, serve as process groups or structure for the model. From the data analysis reported in section 5.8, it can be concluded that all nine panellists accepted the proposed capabilities and did not propose additional capabilities. As shown in the Figure 6-1, TSD, SUP and ORG received the highest rating followed by PM then PD&I. The results were consistent with the analysis conducted on each capability.

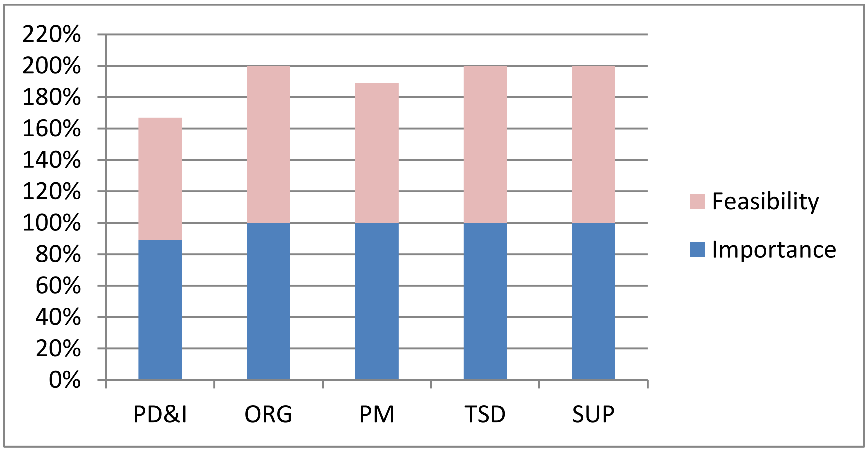


Figure 6-1: Comparison of five capabilities at high level

6.2.2 Evaluation of the Overall Model

In this section, the findings reported in 5.9 are summarised and explained with reference to this study and prior research in this area.

The overall feedback by panellists on the model was very positive; most panellists were very cooperative and provided rich feedback in both rounds. The analysis of quantitative and qualitative data provided shows that all nine-panel members acknowledged the importance of the model as a methodological framework aiming to improve the quality of product development in SSEs and to help in sustaining long-term success. This viewpoint is reported by the literature, stressing that effective internal processes and structures are critical for lowering cost and delivering high quality products or services (Teece, Pisano, et al., 1997).

Despite the observed high consensus for importance for the majority of sub processes included in the model, a few panellists questioned the feasibility of the model and considered the implementation of the entire model an overkill for SSEs given their reported lack of resources, a fact which is highly stressed in the literature (Mishra & Mishra, 2007).

This view made some panellists suggest the need for SSEs to start with the improvement of those processes that are critical for their business. Hence, for this reason there was strong recommendation for classifying the sub processes of the proposed model into must-have (basic) or good to have (advanced). Empirical research studies support this view and reinforce the point. The identification of scope for process improvement through the selection and prioritisation of processes is evident in SPI initiatives (Pino, Garcia, & Piattini, 2009, p. 510); the authors implemented process improvement program in four SSEs within the context of the COPETISOFT project, which aimed to provide Latin American small and medium enterprises with a customised framework for improving their processes. Moreover, A Cater-Steel (2004a, p. 150) in her field experiment of process improvement project (PIP) on 22 SSEs, reported that the assessed organisations were encouraged to select the processes that can provide them with the best benefit.

Therefore, it is important to highlight here that the formalisation of a plan for improvements through the selection and prioritisations of processes is an integral requirement for the adoption of the proposed model, and is critical to its success.

Hence, the improvement initiatives must follow an incremental and iterative approach that allows continuous improvement of practices. This approach is reported by Pino et al. (2008).

From the analysis of results, a number of trends have emerged, which confirm many of the findings of previous research in this area. Following is a discussion of each capability.

6.2.3 Process Definition and Improvement (PD&I)

This capability supports the key principle of quality management and SPI initiatives, which is to define, measure and continually improve an organisation's process infrastructure. Therefore, in the results of the first round, it was a disappointment to see low consensus for importance and feasibility, despite the fact that many panellists considered this critical for the establishment of effective practices that will help organisations to sustain the business. In the second round, results revealed a consensus for both aspects, but with slight disagreement for 1.2 Process Review, which includes activities for the definition of measurements and the collection of process review activities. This is a core process responsible for carrying out continuous improvement of the organisation's process infrastructure.

These findings tend to support the assumption that the institutionalisation and improvement of processes are perceived by the software professionals as relatively difficult to apply in SSES. Available research confirms this view; the systematic review by Pino et al. (2008) about the frequency of improvement efforts in SSEs, reveals that organisations are improving process establishment but their efforts in defining software measurements are insufficient. Hence, in the absence of measurements SPI initiatives are based on informal assessment measures that will result in improvements based on uninformed decisions, which will lead to the failure of SPI efforts.

Moreover, panel members suggested that in order to enhance the feasibility of the model, the defined processes must be simple procedures, where documentation does not exceed 1-2 pages. This view confirms research findings that the adoption of organisational processes must take into account the organisation's size, maturity level, priorities and objectives (C Laporte et al., 2006; Pino et al., 2008).

Additionally, the second round results showed a good level of stability (CV ratings <0.5), which precluded the need for a third round.

Finally, as shown in Appendix D, panellists reached a consensus on the classifications of PD&I sub processes (IQR =0 to 1). *1.1 Process Establishment* was classified as a basic practice, while *Process Review* and *1.3 Process Improvement* were identified as advanced practices. The final version of the model is presented in Appendix E.

6.2.4 **Organisational (ORG)**

As explained in 5.9.2, the ORG processes received high ratings for both importance and feasibility in both rounds. These results show evidence of an increase in awareness among the software community towards the importance of organisational processes. The analysis showed that the customer/supplier relationship received the highest ranking since customers contractually request these practices. On the other hand, panellists perceived the knowledge management (KM) process as infeasible for SSEs, despite the fact they stressed its importance for protecting organisational evolving knowledge; consequently, it was removed from the model.

Another interesting finding is that in the second round the feasibility results for 2.*1 Business Alignment* process show a slight disagreement (IQR =1.5). The analysis of qualitative data revealed that a few panellists hold the opinion that managers of SSEs lack the background and experience to manage the firm strategically. This is a fact strongly highlighted by research, which emphasises that SSEs follow a management style that focuses on operational effectiveness rather than strategic management (Woods & Joyce, 2003). Empirical evidence shows significant association between the firm's performance and its strategic planning and management's strategic approach (Lussier & Halabi, 2010).

As for the classification of ORG processes, panellists collectively identified the entire capability as basic practices. This is considered as further evidence that explains panellists' strong support for ORG processes. The full set of results can be found in Appendix D, section 12.2 on 279.

Moreover, the quantitative results of ORG capability, in both rounds revealed an acceptable degree of stability in the data (CV<0.5). In spite of this, this capability was assessed again in the second round to seek triangulation of data and to assess the model after the removal of *2.5 Knowledge Management*. Therefore, based on these findings the second version of the model has been accepted as the final version for this capability. The final version of the model is presented in Appendix E on page 291

6.2.5 **Technical Solution Development (TSD)**

Quantitative results for this capability revealed high consensus in both rounds where all practices received very high ratings (N=98%-100%), which confirms the view that technical lifecycle processes are followed most by SSEs. The results of a survey conducted on SSEs by A Cater-Steel et al. (2006, p. 143) in Queensland, Australia indicated that primary lifecycle practices relating to requirements and software development exhibit the highest adoption rates in SSEs. These results are confirmed by Pino et al. (2009, p. 512)

However, in the second round the feasibility ratings for some TSD practices such as design, advanced testing and formal documentation of requirements, dropped in rating. The fact that this drop happened in the second round and was just by one level, does not suggest that the panellists are dismissing the importance of those processes, but are giving them a lower priority. Indeed, it is indicative that that in light of the feedback that was provided to panellists in the second round they became more aware of the importance of other processes in the model, yet were still conscious of the resource constraint in SSEs. Hence, this instigated the recommendation by panellists for process selection based on organisation needs or alignment with business objectives.

As for the classification of processes, it was not a surprise that all the processes for TSD were considered basic processes since technical processes have been consistently reported by the literature as being well acknowledged by the software community. The full set of results for TSD is provided in Appendix D, 12.3 on page282.

Finally, the quantitative results for TSD showed high stability (CV ratings <0.5) in both rounds, which reflects panellists' acceptance of the proposed processes and sub processes. Few modifications to the model for this capability occurred in the second round in light of panellists' feedback. The final version of the model is presented in Appendix E on page 291.

6.2.6 Project Management

In the first round, results revealed consensus, showing that panellists perceived PM processes as very highly important, while the feasibility results showed a lack of consensus. The analysis of the qualitative data revealed that the proposed model for this capability is too complex and requires amendments; as a result, dramatic changes were undertaken. The second round results showed consensus with high ratings for almost all the sub processes. A number of processes showed slight disagreement for both importance and feasibility.

From the overall analysis of PM results, we can see that certain trends have emerged. Similar to TSD, PM feasibility results show a drop in panellists' ratings by one level (MED =3). This trend may be because panellists tended to take a more holistic approach in the second round showing more appreciation of capabilities that benefit the entire organisation (PD&I and ORG), together with capabilities related to the project or product (TSD and PM). This importance of organisation level processes for SPI initiatives is emphasised by Pino et al. (2009), who found when conducting process improvements in three SSEs that due to the lack of organisational processes they had difficulty in aligning the improvement proposal with the firm's strategic planning. Therefore, they reported that all three appraised enterprises gave a high degree of priority to the improvement of organisation level processes as it helped them respond fast in the changing business environment.

The second round results for PM showed a good level of stability (CV ratings <0.5), which averted the need for a third round.

Finally, the results in Appendix D, section 12.4 on page 284 show a consensus on the classification of all PM sub processes as basic practices. Hence, panellists expressed support for the final version of PM. The full model for PM capability is presented in Appendix E on page 291.

6.2.7 Support

In the first round, panellists expressed collective agreement on the very high importance of SUP processes and considered them critical for maintaining the product value to customers. Despite this highlighted importance, the first round feasibility results showed a lack of consensus for all the processes (especially configuration and change management). Panellists felt that support processes are not assigned to dedicated teams. This view is challenged by Rashid et al. (2009) who reported the need for effective formal managerial processes for maintaining the software during post deployment stages. In the second round, we note a change in opinion regarding the feasibility of the SUP capability, where a consensus was reached for almost the majority of processes. Additionally, a few panellists rated low the processes that involve proactive problem identification, review and assessment.

Moreover, the analysis of results revealed stability in data, which made one more round unnecessary and confirmed that panellists approved the SUP capability.

Furthermore, results reveal consensus for the classification of SUP sub processes being basic processes, except two sub processes related to configuration management. Considering the low consensus for configuration management in the first round and the results of classification, it may suggest the entire configuration management process should be considered an advanced practice. The final version of the model is presented in Appendix E.

6.3 Summary

The discussion in this Chapter presented panellists ratings and views regarding the processes and sub processes for each capability. Important findings identified in the literature were compared to panellists' views concerning the processes and sub processes. Additionally, the discussion included explanation and justifications for the identified DCs and explanation of panellists' classifications of the processes.

The discussion highlighted that the establishment of processes is perceived as a feasible practice for SSEs. However, processes related to the review and improvement of processes received lower feasibility ratings. Moreover, panellists showed high appreciation for the majority of processes that belong to ORG capability, particularly those related to building customer relationship management. The discussion also underlined the significance of following agile approaches in testing and design. Finally, the SUP processes that involve proactive problem resolution, problem closure and review, and configuration audits were viewed by panel members as less feasible for SSEs.

The next chapter concludes this research study, discusses the limitations of the study and its contribution to the theory and practice.

CHAPTER 7 - CONCLUSION

7.1 Introduction

This chapter concludes the research study by providing a summary of the preceding work and presenting the conclusion related to the research problem. The contribution of the research to the body of knowledge is discussed, as well as the implication for theory and the limitations of the study. The last paragraph determines areas of future research. Following is a summary of the research chapters, stating the findings of each chapter and its contribution to answering the research questions.

7.2 Research Motivation

The first chapter discussed the problem and motivation behind this research project. This study was motivated by the need for tailored SPI models for SSEs that can replace international standards, which are perceived by the IT community as too complex for application in SSEs. The literature review in Chapter 2 identified the absence of a holistic multifaceted single model that will guide SSEs in their process improvement initiatives. Moreover, the inclusion of DCs is a unique characteristic of the model proposed by this study, for helping SSEs in addressing business challenges within dynamic markets. The overarching research question was stated in Chapter 1:

> *What do experts in software development projects construe as relevant practices and Dynamic Capabilities that make up a holistic multi-facet software process improvement model?*

7.3 Responding to Research Questions

In this section, a summary of the chapters that contributed to answering the research questions is provided. This is summarized in Figure 7-1.

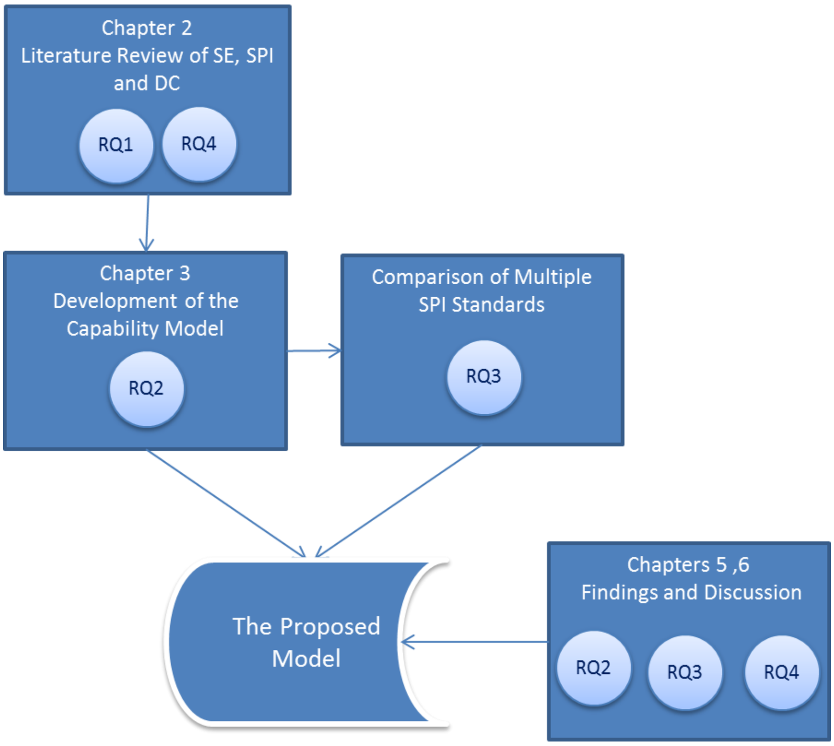


Figure 7-1: How the research method addressed the questions

Chapter 2 reviewed statistics and research related to SEs worldwide and identified a set of common characteristics that can be generalised, and are considered factors that can impact the adoption of process reference models in SSEs. A summary of the findings is provided in Appendix A.

Moreover, Chapter 2 also reviewed the literature related to SPI. The review revealed barriers hindering SPI efforts including a set of inherent characteristics that can be generalised to SSEs (refer to Appendix A, Table A-3). Both areas of research (SE and SSE) revealed common challenges facing small firms in general regardless of the business context (refer to the comparison in Appendix A,

Table A-4).

Furthermore, a critical review of dynamic capabilities was conducted. The findings relating to DCs revealed that this theory has not been explored systematically in SPI literature. The review highlighted the difference between operational and dynamic capabilities and provided insights on applying the sense and response model in developing DC processes that enable the firm to employ sensing processes to detect potential internal and external threats and changes in the business environment.

Chapter 2 literature review answers fully research question RQ1 and contributes to answering RQ4.

In Chapter 3, a critical analysis of existing research was performed to understand and explore the activities performed by firms through the lifecycle of software products. The reviewed literature revealed that for successful development of software products it is necessary for SSEs to build five capabilities. These identified capabilities laid the foundation and structure for the proposed model. This research milestone contributed to answering RQ2.

Further, a detailed analysis of the standards that were selected for the study as discussed in Chapter 1 (section 11.8) was performed to understand the processes and methodological approach proposed by each standard as a guideline for organisations seeking to adopt best practice. Following this analysis, a thorough comparison of the selected standards was carried out resulting in a set of identified processes and sub processes for each of the five capabilities. The identified capabilities and underlying processes constituted the proposed model. Moreover, the review of DC literature in Chapter 2 resulted in the identification of the 'sense and respond' model and the categorization of processes and sub processes into operational and DCs. Following this step, each DC process was categorised into sensing, seizing or transforming process. The outcome of this stage resulted in the development of the proposed model. This fulfils the partial study goals for RQ2, RQ3 and RQ4.

Finally, in order to achieve the study goals, the model was empirically assessed using the Delphi technique. The analysis of the data collected during the two Delphi rounds and the discussion of findings in Chapters 5 and 6 highlighted some important implications for process adoption and capability building in SSEs.

The study makes an account of an array of technical and organisational capabilities associated with the development of software in the small firm context. The findings revealed that panel members gave higher priority to technical and project level capabilities compared to organisational capabilities. Additionally, a very high rating was given to ORG, reflecting the significance of this capability. This is recognised as a promising response reflecting higher awareness among the software community regarding organisational issues. Research reports that organisational issues are more of a problem for high maturity firms, and that project and technical problems are the concern of low maturity firms (A Cater-Steel et al., 2006)

The participating panel members provided their subjective views as experts in the field; thus their conveyed opinions offer an overview about the state-of-the-art knowledge of SPI programmes used in SSEs and barriers to their adoption, in addition to providing recommendations on how to pursue successful process adoption and improvement initiatives in the small software firm context.

7.4 Contribution to Theory and Practice

Contribution #1

The major contribution of this research is the development and empirical assessment of the proposed model. The model isinformed by accepted practice and the application of DC theory. As discussed in section 6.2.2, the participating panel members have acknowledged that the model is comprehensive, and if applied it can help SSEs to survive and further improve their software products and overall business. Panellists' continuous consideration of SSEs limited resources flags an important reality, which is why SSEs make limited use of best practice. However, empirical findings report that SSEs are successful in achieving notably higher capability levels as a result of SPI initiatives (A Cater-Steel et al., 2006).

Therefore, in this study it is perceived that the model can help small businesses sustain long-term benefits. The literature review in Chapter 2 presented statistics as well as reports from the literature stressing that small business survival rates are very low (Lussier & Halabi, 2010). Table 7-1 provides a list of suggested benefits from employing the proposed model of this study, which are well reflected by Lussier's success and failure model, as success factors (Lussier & Halabi, 2010). Lussier's model is presented in Appendix A. However, this perceived benefit requires empirical assessment to assess the extent of benefits and confirm the feasibility of the model for SSEs.

Further to the benefits discussed above, this study presents a model that can provide a holistic approach to tackling the software development process as well as organisational performance. This is presented in Table 7-1. It has been highlighted by Pryor et al. (2010) that generally quality initiatives fail in managing the business from a strategic and systems perspective.

Table 7-1: Proposed model perceived benefits

PD&I

- Document and improve organisational processes

ORG

- Improve firm's strategic planning.
- Improves the selection and retention of talented employees
- Enhance enterprise learning and knowledge dissemination
- Adequate financial control

PM

- Enhance project outcomes by meeting time, cost and budget
- Higher consistency in the quality of software products
- Proactive identification of problems at project and organisational level

TSD

- Better understanding of customer requirements
- Ensures that products meet customer requirements
- Improved product quality

SUP

- Improved customer services post-delivery of the product

Application of DCs

- Enhance strategic planning
- Creates coherence between the internal and external environment
- Allows higher adaptability and responsiveness to changing market conditions

Contribution #2

As well as the developed model that is intended for small enterprises, this study contributes to the literature on small businesses by reporting how software development practitioners view the challenges facing SSEs and comparing it to academic research in the area. The findings show that both parties hold a similar view of the problem. Moreover, the study provides a number of insights, and recommendations provided by panel members, that can contribute to improving quality initiatives in SSEs.

For example, the analysis of findings in the second round showed higher acknowledgements for capabilities that are not related to product development. This is attributed to the increase in awareness about the benefits of these capabilities.

We have also observed from the analysis of the results that the lack of available resources is seen to compromise the success of any SPI efforts, but, the literature reports that it is possible to undertake an SPI initiative and to make real progress without big investment (C Laporte et al., 2005). Following is a list of recommendations to guide enterprises in their improvement efforts:

- A gradual approach to the adoption of the five capabilities based on continual assessment of the improvement cycle. Firms must start with the processes that are most needed by the organisation or are most aligned with their business objectives.
- The adoption of agile development practices to enhance the feasibility of the model.
- The increase in employee awareness towards the importance of process oriented quality initiatives can contribute to the success of SPI initiatives.
- Firms' motivation must be to improve their processes in order to make them more effective and efficient, rather than to gain certifications.
- Firms must seek external guidance and mentoring for successful adoption of the proposed model.

Contribution #3

As far as is known, this study is the only study that has explored the adoption of DCs in a SPI process improvement model. The inclusion of DCs in the model can improve the competitive advantage of SSEs. However, DCs have been treated somewhat narrowly in this study. This is one of the limitations as discussed in section 0 below.

Contribution #4

In this study, the Delphi technique has been used as the methodology for collecting data. Delphi has been widely used and explored in the literature. Nevertheless, this study provides detailed information and tips, and presents successful example of the application of Delphi in this field. The Delphi method helped in receiving rich data from participants who provided written qualitative data. The fact that they documented the comments themselves implies more accuracy in the provided data.

To conclude, this study contributes to the theory by improving the understanding about the adoption of software development processes in the small enterprise context, as well as contributing to a better understanding of the strategic management theory “Dynamic Capabilities” and its application in the software development context. A set of necessary processes and DCs has been collected in a comprehensive holistic capability model. The developed model has been empirically assessed and can be examined further by academics and practitioners.

Moreover, the study will benefit all stakeholders in the software industry including developers, owners, customers, suppliers and researchers in this field.

Contribution #5

A valuable contribution of this study is the use of online social media to recruit research participants. In this study, announcements were published on number of social media venues, precisely professional groups at LinkedIn forum to recruit participants from different geographical locations. The outcome was very successful and helped in overcoming many of the challenges including geographical barriers, insufficient time and financial resources.

7.5 Limitations

Reflecting on the research design and execution, the following limitations have been identified.

The first limitation concerns the fact that each Delphi round provides a viewpoint limited to the sample of panel members. As discussed in section 4.5.12 on page 91, some measures have been taken to enhance the study rigour and validity. However, if it had been possible to have larger panel size, other members would have had different viewpoints that could enhance the richness and validity of the data. The time constraint for this study did not allow for recruiting more panel members.

The second limitation of the study concerns enhancing the validity of the research through triangulation of data sources. This research only used a Delphi surveys to collect data. To enhance the richness of data, other sources of data collection could have been used, such as semi structured interviews or surveys.

Another research limitation concerns DCs. The design of the questionnaire limited the data collected for DCs to qualitative data. Collecting quantitative data would have required panellists to spend more time filling in the surveys, which could have resulted in an even lower panel size. A separate round for DCs would have helped in creating more awareness about its benefits and in collecting richer data.

7.6 Future Research Direction

This study found that practitioners supported the assumption that the adoption of the proposed model can improve the maturity of the organisation and can positively contribute to a positive significant relationship with performance. However, further empirical evidence is required to support this claim.

A number of SSE funded software improvement programmes initiated by governments or government organisations have been reported by several studies. According to Pino et al. (2008) countries that have reported improvement efforts in SSES most are Australia, Ireland, Finland, Denmark, Brazil, USA, Sweden and Mexico. It is worth noting here that to the best of our knowledge, there has been no initiatives by the New Zealand government with respect to SPI programmes in SSEs. Since 97% of all public and private enterprises in New Zealand are predominantly small in size ("SMEs in New Zealand: Structure and Dynamics 2011," 2011), there is a need for a call for national efforts into research on improving processes in small New Zealand software development firms, led and funded by the government, universities and the private sector. If such an initiative ever comes about, then it would be a good opportunity for a longitudinal study to implement the proposed model in a number of New Zealand small firms and assess its impact empirically.

Further research is required to explore the application of DCs in SPI models. This study provides an example of the application of DCs, albeit narrowly. Therefore, there is a need for future research to assess empirically the benefit and usefulness of the suggested DCs on the proposed model. There is a need for research to suggest different application of DC theory in SPI initiatives.

Reference List

(OGC)., O. o. G. C. (2010). *The Introduction to the ITIL Service Lifecycle Book (Official Introduction)*: The Stationery Office, OGC.

Abdelrahim, A., & MBA, M. (2007). Critical analysis and modelling of small business performance (Case Study: Syria). Journal of Asia Entrepreneurship and sustainability, 3(2). *Journal of Asia Entrepreneurship and Sustainability, 3*(2).

Abran, A., Boruque, P., & Trip, L. L. (2004). *SWEBOK: Guide to Software Engineering Body of Knowledge*. California, Los Alamitos: IEEE Computer Socieity.

Ahmed, R. E. (2006). Software maintenance outsourcing: Issues and strategies. *Science Direct*(32), 449–453.

Alexandre, S., Renault, A., & Habra, N. (2006). *OWPL: A Gradual Approach for Software Process Improvement In SMEs.* Paper presented at the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications.

Allison, I., & Merali, Y. (2007). Software Process Improvement as Emergent Change:a Structural Analysis. *Information and Software Technology, 49*(6), 668-681.

Barafort, B., Di Renzo, B., & Merlan, O. (2002). *Benefits resulting from the combined use of ISO/IEC 15504 with the Information Technology Infrastructure Library (ITIL)*, Rovaniemi, Finland.

Barkham, R., Gudgin, G., & Hanvey, E. (1996). *Determinants of Small Firm Growth: An Inter-Regional Study in the United Kingdom 1986-90* (Vol. 12): Psychology Press.

Barreto, I. (2010). Dynamic capabilities: A review of past research and an agenda for the future. *Journal of Management, 36*(1), 256-280.

. Capability Maturity Model Integration for Development (CMMI-DEV) Version 1.3. (2010). Pittsburgh, Pennsylvania: Carnegie Mellon University Software Engineering Institute

Cater-Steel, A. (2001). *Process improvement in four small software companies.* Paper presented at the Software Engineering Conference, 2001. Proceedings. 2001 Australian.

Cater-Steel, A. (2004a). *An evaluation of software development practice and assessment-based process improvement in small software development firms.* Griffith University.

Cater-Steel, A. (2004b). *Low-rigour, rapid software process assessments for small software development firms.* Paper presented at the Software Engineering Conference, 2004. Proceedings. 2004 Australian.

Cater-Steel, A., Tan, W., & Toleman, M. (2006). *Challenge of adopting multiple process improvement frameworks.* Paper presented at the Proceedings of 14th European conference on information systems (ECIS 2006).

Cater-Steel, A., Toleman, M., & Rout, T. (2006). Process improvement for small firms: An evaluation of the RAPID assessment-based method. *Information and Software Technology, 48*(5), 323-334.

Cattaneo, F., Fuggetta, A., & Sciuto, D. (2001). Pursuing Coherence in Software Process Assessment and Improvement. *Software Process Improvement and Practice*, 3-22.

Creswell, J. (2012). *Qualitative inquiry and research design: Choosing among five approaches*. London: Sage.

Creswell, J. W. (2008). Research design: Qualitative, quantitative, and mixed methods approaches. *Sage Publications, Incorporated*.

Crotty, M. (1998). *The foundations of social research: Meaning and perspective in the research process*: Sage Publications Limited.

Dajani, J. S., Sincoff, M. Z., & Talley, W. K. (1979). Stability and agreement criteria for the termination of Delphi studies. *Technological Forecasting and Social Change, 13*(1), 83-90.

Dalcher, D. (2009). Attaining and sustaining organisational excellence: process improvement as a moving target. *Software Process: Improvement and Practice, 14*(2), 63-64.

Dangol, R. (2012). Linking dynamic capabilities, operational capabilities and firm resources to performance.

Davis, C. H., & Sun, E. (2006). Business Development Capabilities in Information Technology SMEs in Regional Economy. *The Journal of Technology Transfer, 31*(1), 145-161.

Day, J., & Bobeva, M. (2005). A generic toolkit for the successful management of Delphi studies. *The Electronic Journal of Business Research Methodology, 3*(2), 103-116.

Easterbrook, S., Singer, J., Storey, M., & Damian, D. (2008). Selecting Empirical Methods for Software Engineering Research *Guide to Advanced Empirical Software Engineering* (pp. 285-311). London: Springer.

Ehsan, N., Malik, O. A., Shabbir, F., Mirza, E., & Bhatti, M. W. (2010). *Comparative Study for PMBOK and CMMI frameworks and identifying possibilities for integrating ITIL for addressing the needs of IT service industry.* Paper presented at the Proceedings of the 2010 IEEE ICMIT.

Eisenhardt, K. M., & Martin, J. A. (2000). Dynamic capabilities: what are they?. *Strategic management journal, 21*(10-11), 1105-1121.

Fening, F., Pesakovic, G., & Amaria, P. (2008). Relationship between quality management practices and the performance of small and medium size enterprises (SMEs) in Ghana. *International Jouranal of Quality & Reliability Management, 25*.

Flower, P. (2009). Research Philosophies–Importance and Relevance. *Economic Record, 1*(3).

Galinac, T. (2009). Empirical evaluation of selected best practices in implementation of software process improvement. *Information and Software Technology, 51*, 1351-1364.

Goldenson, D. R., & Herbsleb, J. D. (1995). After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success. Pittsburg: Software Engineering Institute.

Habra, N., Alexandre, S., Desharnais, J., Laporte, C., & Renault, A. (2008). Initiating software process improvement in very small enterprises: Experience with a light assessment tool. *Information and Software Technology, 50*(7), 763-771.

Haeckel, S. H. (2000). The premise and promise of sense and respond. *IBM Corporation*.

Hallowell, M. R., & Gambatese, J. A. (2009). Qualitative research: Application of the Delphi method to CEM research. *Journal of construction engineering and management, 136*(1), 99-107.

Hanafin, S., & Brooks, A. M. (2005). The Delphi technique: A methodology to support the development of a national set of child well-being indicators. *Stationery Office*.

Hansen, B., Rose, J., & Tjornehoj, G. (2004). Prescription, description, reflection: the shape of the software process improvement field. *International Journal of Information Management, 24*, 457-472.

Hasson, F., & Keeney, S. (2011). Enhancing rigour in the Delphi technique research. *Technological Forecasting and Social Change, 78*(9), 1695-1704.

Helfat, C., Finkelstein, S., Mitchell, W., Peteraf, M., Singh, H., Teece, D., & Winter, S. (2009). *Dynamic Capabilities: Understanding strategic change in organizations*: Wiley. com.

Helfat, C., & Peteraf, M. (2003). The dynamic resource-based view: Capability lifecycles. *Strategic management journal, 24*(10), 997-1010.

Higgs, J., Horsfall, D., & Grace, S. (2009). *Writing qualitative research on practice*: Sense Publishers.

Hochstein, A., Zarnekow, R., & Brenner, W. (2005). *ITIL as common practice reference model for IT service management: formal assessment and implications for practice.* Paper presented at the e-Technology, e-Commerce and e-Service, 2005. EEE'05. Proceedings. The 2005 IEEE International Conference on.

Holden, M. T., & Lynch, P. (2004). Choosing the appropriate methodology: understanding research philosophy. *The marketing review, 4*(4), 397-409.

. ISO/IEC 12207:2008 (2008) *Systems and software engineering — Software life cycle processes*. Geneva, Switzerland.

. ISO/IEC 15504-5:2012. (2012) *Information technology — Process assessment — Part 5: An exemplar software life cycle process assessment model*. Geneva, Switzerland

. ISO/IEC/IEEE 15288:2015. (2015) *Systems and software engineering — System life cycle processes*. Geneva, Switzerland

Kelly, D. P., & Culleton, B. (1999). Process improvement for small organizations. *Computer, 32*(10), 41-47.

Kirk, D., & MacDonell, S. (2009). *A System Approach to Software Process Improvement in Small Organisations.* Paper presented at the EuropSPI.

Kivelä, M. (2007). Dynamic capabilities in small software firms. Helsinki School of Economics.

Lam, W., & Shankararaman, V. (1998). *Managing Change in Software Development Using a Process Improvement Approach.* Paper presented at the IEEE Euromicro Conference, 1998. Proceedings. 24th.

Laporte, C., Alain, A., & Renault, A. (2006). *Applying ISO/IEC Software Engineering Standards in Small Setting: Historical Perspective and Initial Achievements.* Paper presented at the Proceedings of SPICE 2006 Convference, Luxembourg.

Laporte, C., Renault, A., Desharnais, J., Habra, N., Abou El Fattah, M., & Bamba, J. (2005). *Initiating software process improvement in small enterprises: Experiment with micro-evaluation framework.* Paper presented at the SWDC-REK, International Conference on Software Development, University of Iceland, Reykjavik, Iceland.

Lester, N. G., Wilkie, F. G., McFall, D., & Ware, M. P. (2010). Investigating the role of CMMI with expanding company size for small-to medium-sized enterprises. *Journal of Software Maintenance and Evolution: Research and Practice, 22*(1), 17-31.

Lussier, R. (1995). A Nonfinancial Business Success Versus Failure Prediction Model for Young Firms. *Journal of Small Business Management, 33*(1), 8-20.

Lussier, R., & Corman, J. (1996). A BUSINESS SUCCESS VERSUS FAILURE PREDICTION MODEL FOR ENTREPRENEURS WITH 0-10 EMPLOYEES. *Journal of Small Business Strategy, 7*(1), 21-35.

Lussier, R., & Halabi, C. (2010). A Three-Country Comparison of the Business Success Versus Failure Prediction Model. *Journal of Small Business Management, 48*(3), 360-377.

Lussier, R., & Pfeifer, S. (2000). A Comparison of Business Success versus Failure Variables between U.S. and Central Eastern Europe Croatian Entrepreneurs. *Entrepreneurship: Theory and Practice*.

Marrone, M., & Kolbe, L. M. (2011). Uncovering ITIL claims: IT executives' perception on benefits and business-IT alignment. *Information Systems and eBusiness Management, 9*(3), 363-380.

Marrone, M., Gacenga, F., Cater-Steel, A., & Kolbe, L. (2014). IT Service Management: a Cross-National Study of ITIL Adoption. *Communications of the Association for Information Systems 34*(1), 865-892.

Mathiassen, L., & Vainio, A. M. (2007). Dynamic capabilities in small software firms: A sense-and-respond approach. *IEEE Transactions on Engineering Management,,, 54*(3), 522-538.

Mazzarol, T., Reboud, S., & Soutar, G. N. (2009). Strategic planning in growth oriented small firms. *International Journal of Entrepreneurial Behaviour & Research, 15*(4), 320-345.

McHugh, O., & Hogan, M. (2011). Investigating the rational for adopting an internationally-recognised project management methodology in Ireland: The review of the project manager. *International Journal of Project Management*, 637-646.

Menken, I., Blokdijk, G., & Malone, T. (2009). *ITIL V3 MALC - Managing Across the Lifecycle of IT Services Best Practices Study and Implementation Guide*: Emereo Pty Ltd.

Miller, P. (2007). *An SEI process improvement path to software quality.* Paper presented at the Quality of Information and Communications Technology, 2007. QUATIC 2007. 6th International Conference on the.

Mishra, D., & Mishra, A. (2007). Efficient software reveiw process for small and medium enterprises. *The Institution of Engineering and Technology, 1*(4), 132-142.

Nasir, M., Ahmad, R., & Hassan, N. (2008). Resistance Factors in the Implementation of Software Process Improvement Projects in Malaysia. *Journal of Computer Science, 4*(3), 211-219.

Niazi, M. (2009). Software Process Improvement Implementation: Avoiding Critical Barriers. *The Journal of Defense Software Engineering*, 24-27.

Niazi, M., Babar, M., & Katugampola, N. (2008). Demotivators of Software Process Improvement:An Empirical Investigation. *Software Process Improvement and Practice, 13*, 249-264.

Nielsen, P. A., & Kautz, K. (2008). *Software Processes & Knowledge: Beyond Conventional Software Process Improvement*. Aalborg: Software Innovative Publisher.

Niessink, F., & Van Vliet, H. (2000). Software Maintenance from a Service Perspective. *Jorunal of Software Maintenance: Research and Practice, 12*(2), 103-120.

OECD. (2010). Small and Medium-Sized Enterprises. Paris.

Okoli, C., & Pawlowski, S. D. (2004). The Delphi method as a research tool: an example, design considerations and applications. *Information & Management, 42*(1), 15-29.

Patnayakuni, R., & Ruppel, C. P. (2010). A socio-technical approach to improving the systems development process. *Inf Syst Front*(12), 219–234.

Peldzius, S., & Ragaisis, S. (2011). *Comparison of maturity levels in CMMI-DEV and ISO/IEC 15504.* Paper presented at the American conference on applied mathematics and the 5th WSEAS international conference on Computer engineering and applications

Pino, F., Baldassarre, M., Piattini, M., & Visaggio, G. (2010). Harmonizing maturity levels from CMMI-DEV and ISO/IEC 15504. *Journal of Software Maintenance and Evolution: Research and Practice*, 22:279-296.

Pino, F., Garcia, F., & Piattini, M. (2009). *Key processes to start software process improvement in small companies.* Paper presented at the Proceedings of the 2009 ACM symposium on Applied Computing.

Pino, F., García, F., & Piattini, M. (2008). Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal, 16*(2), 237-261.

Pino, F., Pardo, C., García, F., & Piattini, M. (2010). Assessment methodology for software process improvement in small organizations. *Information and Software Technology, 52*(10), 1044-1061.

Pino, F., Pedreira, O., Garcia, F., Rodriguez, L., & Piattini, M. (2010). Using Scrum to guide the execution of software process improvement. *The Journal of Systems and software, 83*, 1662-1667.

Pleatsikas, C., & Teece, D. (2001). The analysis of market definition and market power in the context of rapid innovation. *International Journal of Industrial Organization, 19*(5), 665-693.

Porter, M. E. (1979). How competitive forces shape strategy.

Priem, R. L., & Butler, J. E. (2001). Is the resource-based "view" a useful perspective for strategic management research? *Academy of management review, 26*(1), 22-40.

Pryor, M. G., Toombs, L., Anderson, D., & White, J. C. (2010). What management and quality theories are best for small businesses. *Journal of Management and Marketing Research, 3*(1), 1-12.

Ragaisis, S., Peldzius, S., & Simena, J. (2011). *Comparison of maturity levels in CMMI-DEV and ISO/IEC 15504.* Paper presented at the Proceedings of the 2011 American conference on applied mathematics and the 5th WSEAS international conference on Computer engineering and applications, Puerto Morelos, Mexico.

Rashid, A., Wang, W. Y., & Dorner, D. (2009). *Gauging the differences between expectation and systems support: the managerial approach of adaptive and perfective software maintenance.* Paper presented at the Cooperation and Promotion of Information Resources in Science and Technology, 2009. COINFO'09. Fourth International Conference on.

Rayens, M. K., & Hahn, E. J. (2000). Building consensus using the policy Delphi method. *Policy, politics, & nursing practice, 1* (4), 308-315.

Rose, K. H. (2013). A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition. *Project Management Journal, 44*(3).

Rousseva, R. (2007). *Approach for Analysing Capabilities in Latecomer Software Companies*.

Santos, G., Montoni, M., Vasconcellos, J., Figueiredo, S., Cabral, R., Cerdeiral, C., . . . Rocha, A. R. (2007). *Implementing software process improvement initiatives in small and medium-size enterprises in Brazil.* Paper presented at the Quality of Information and Communications Technology, 2007. QUATIC 2007. 6th International Conference on the.

Schmidt, R., Lyytinen, K., Kei, M., & Cule, p. (2001). Identifying software project risks: an international Delphi study. *Journal of management information systems, 17*(4), 5-36.

Schmidt, T., & Mathiassen, L. (2009). Agility in a small software firm: a sense-and-respond analysis. *International Journal of Business Information Systems, 4*(1), 85-104.

Shenton, A. K. (2004). Strategies for ensuring trustworthiness in qualitative research projects. *Education for information, 22*(2), 63-75.

. SMEs in New Zealand: Structure and Dynamics 2011. (2011): Ministry of Economic Development.

Teece, D. (2007). Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. *Strategic management journal, 28*(13), 1319-1350.

Teece, D., Gary, P., & Shuen, A. (1997). Dynamic Capabilities and Strategic Management. *Strategic management journal, 18*(7), 509-533.

Teece, D., & Pisano, G. (1994). The dynamic capabilities of firms: an introduction. *Industrial and Corporate Change, 3*(3), 537-556.

Teece, D., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic management journal, 18*(7), 509-533.

Tschang, T. (2001). The Basic Characteristics of Skills and Organisational Capabilities in the Indian Software Industry.

Turner, R., Ledwith, A., & Kelly, J. (2010). Project management in small to medium-sized enterprises: Matching processes to the nature of the firm. *International Journal of Project Management, 28*(8), 744-755.

Van Zolingen, S. J., & Klaassen, C. A. (2003). Selection processes in a Delphi study about key qualifications in Senior Secondary Vocational Education. *Technological Forecasting and Social Change, 70*(4), 317-340.

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS quarterly, 37* (1), 21-54.

Von Der Gracht, H. A. (2012). Consensus measurement in Delphi studies: review and implications for future quality assurance. *Technological Forecasting and Social Change*(79), 1525-1536.

Wangenheim, C., da Silva, D., Buglione, L., Scheidt, R., & Prikladnicki, R. (2010). Best practice fusion of CMMI-DEV v1.2 (PP, PMC, SAM) and PMBOK 2008. *Information and Software Technology*, 749-757.

Wiegers. (1993). Implementing Software Engineering in a Small Workgroup. *Computer Language*.

Wilkie, F. G., McFall, D., & McCaffery, F. (2005). An evaluation of CMMI process areas for small-to medium-sized software development organisations. *Software Process: Improvement and Practice, 10*(2), 189-201.

Wood, M. S., & Michalisin, M. D. (2010). Entrepreneurial Drive in the Top Management Team: Effects on Strategic Choice and Firm Performance. *Journal of Leadership & Organizational Studies, 17*(3), 222–239.

Woods, A., & Joyce, P. (2003). Owner-Managers and the Practice of Strategic Management. *International Small Business Journal, 21*, Vol 21(22): 181–195.

Yu-Chih Liu, J., Chen, V., Chan, C., & Lie, T. (2008). The impact of software process standardization on software flexibility and project management performance:Control theory perspective. *Information and Software Technology-Elsevier, 50*, 889–896.

Zavala-Ruiz, J. (2008). Organizational Analysis of Small Software Organizations: Framework and Case Study In H. Oktaba & M. Piattini (Eds.), *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies* (pp. 1-41). Hershey, PA: Information Science Reference.

Zazworka, N., Basili, V., & Shull, F. (2010). *Tool Supported Detection and Judgement of Nonconformance in Process Execution.* Paper presented at the Third Internation Symposium on Empirical Software Engineering and Measurement.

Zollo, M., & Winter, S. G. (1999). From organizational routines to dynamic capabilities. *INSEAD*.

Appendix A - Summaries of Literature Review Findings

Table A-1- Summary statistical reports findings

The number of enterprises with less than 20 employees exceeds 70% in many countries. The software industry in most countries is mainly composed of small enterprises. The distribution of SEs is primarily towards the younger age groups. Employee turnover rates tend to increase as firm size decreases. SEs had the greatest proportion of part-time staff. SEs are less likely to survive than larger firms. SEs are likely to change size and become either smaller or larger

Table A-2 – Literature review findings on characteristics of Small Enterprises (SEs)

Zavala-Ruiz (2008)	• Structure, management and process are different from LEs. • Economic performance is not inferior to LEs.
Woods & Joyce (2003)	• Have limited human resources, loose division of labour, minimal differentiation among its business units, flat structure and non-formalised administrative behaviour.
Fening, Pesakovic, & Amaria, (2008)	• Lack of strategic planning processes.
Pryor et al. (2010)	• Unlikely to utilize strategic management models and long term planning. • Entrepreneurs who develop small businesses usually have little interest to establish processes.
Mazzarol et al. (2009)	• Managers at SEs lack the background and expertise to manage the business strategically.
Zavala-Ruiz (2008)	• The unique and multiple roles played by the owner-manager degrade the decision-making process and leads to inappropriate actions

Table A-3 – Literature review findings of characteristics of Small Software Enterprises (SSEs)

A Cater-Steel (2001)	• Low process maturity and more attention is given to project and technical issues than organisational issues. • Concentrate on day-to-day issues and do not have resources available for SPI. • Shortage of IT staff and the loss of key ones is difficult to replace • Lack of resources is an obstacle for staff training. • Processes exist with varying capability levels; some practices can be very mature while others are extremely poor
C Laporte et al. (2005)	• Lack of formal processes particularly poor planning process or does not exist. • No staff training and success was highly dependent on individual skills.

A Cater-Steel et al. (2006)	• The involvement of owner-manager in all aspects of the business
Zavala-Ruiz (2008)	• Chaotic administrative and operational management characterised by a culture of urgency.
Alexandre et al. (2006).	• Most business focus is on arising business matters. • The lack of resources exists at all levels including monetary, human, time and in particular, the quality processes suffer the most • Remarkable competence on the technical domain, and low attention to organisational and quality processes • Simplified Life cycle model • Learning and knowledge management is rarely observed.
F Pino, O Pedreira, et al. (2010)	• Flat structure with free management style • Lightweight processes • Constant communication between project members and customers.

Table A-4 – Comparison between SEs and SSEs

Characteristics of SEs	Characteristics of SSEs
• Flatter structure with non-formalised administrative behaviour and informal communications	• Flat structure with free management style
• The owner-managers' entrepreneurial characteristics have an impact on the development of the business such as competence, strategic planning, entrepreneurial orientation and resource management skills	• The role of the owner-manager to commit and lead the SPI initiative is critical for its success.
• Small Enterprises management style focuses on day-to-day operation rather than strategic management. • Strategic management is planning for future expansion in products, services and markets, which require risk management, and maximising the utilisation of the limited resources.	• The organisation focus is on the technical engineering aspect and ignores other organisational influencing factors (e.g. environmental, culture, employee development) • Process oriented and disciplined practices will likely to promote strategic and operational planning.
• SEs have limited human resources, loose division of labour, minimal differentiation among its business units.	• The lack of resources assigned to the software process exists at all levels including monetary, human, time and in particular, the quality processes
• There is minimal training of staff. There is a need for educating and training managers on the use strategic management practices and tools. •	• Training and professional development practices are very limited due to the severe budget constraint.

• The need to create a learning environment	• Learning and knowledge management can be rarely observed. There is a need to harness and focus the capability of employees by creating a learning environment.
• Enterprise success is linked to the behaviour and attributes associated with the enterprise people.	• Properly assigning people to development roles is crucial for creating productive teams.
• There is evidence that rapidly growing firms are linked to the entrepreneurial orientation of the organisation or entrepreneurial drive of the individuals in the organisation	• Business development capabilities and having a strategic direction are critical to the success and growth of SSEs.

Appendix B - Comparison of Models

Category	SWEBOK Knowledge Areas	CMMI – DEV	CMMI Acquisition	ISO/IEC 15504	ISO/IEC 12207	PMBOK	ITIL
Process Definition and Improvement	Software Engineering Process • Process Implementation and Change • Process Definition	**OPD,** SG1 SP1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7 **OPF** SG1, SP1.1, 1.2, 1.3 SG2, SP2.1,2.2, SG3, SP3.1, 3.2, 3.3., 3.4 **GG1**, GP1.1, **GG2,** GP2.1, 2.2, 2.3, 2.4,2.5, 2.6, 2.7, 2.8, 2.9 GP2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10 **GG3, GP** 3.1, 3.2	**OPD,** SG1 SP1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7 **OPF** SG1, SP1.1, 1.2, 1.3 SG2, SP2.1,2.2, SG3, SP3.1, 3.2, 3.3., 3.4 **GG1**, GP1.1, **GG2,** GP2.1, 2.2, 2.3, 2.4,2.5, 2.6, 2.7, 2.8, 2.9 GP2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10 **GG3, GP** 3.1, 3.2	**PA1.1**, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2 5.7 Process Improvement (PIM) **5.7.1 PMI.1,** BP1, BP2, BP3, BP4, BP5,BP6, **5.7.2 PMI.2,** BP1, BP2, BP3, BP4, BP5,BP6, BP7,BP8, **5.7.3 PMI.3,** BP1, BP2, BP3, BP4, BP5,BP6, BP7,BP8,BP9	6.2 Organizational Project-Enabling Processes, 6.2.1, 6.2.1.3.1, 6.2.1.3.2, 6.2.1.3.3,	None	None
Performance Measurement and Continuous Improvement	Software Engineering Process • Process Assessment • Process and Product Measurement	Decision Making **CAR.** SG1, SP1.1, 1.2 SG2, SP2.1, 2.2,2.3 **DAR** SG1, SP1.1, 1.2, 1.3, 1.4, 1.5, 1.6 Quantitative	Decision Making **CAR.** SG1, SP1.1, 1.2 SG2, SP2.1, 2.2,2.3 **DAR** SG1, SP1.1, 1.2, 1.3, 1.4, 1.5, 1.6 Quantitative	**MAN.4**, BP1, BP2, BP3, BP4, BP5,BP6, BP7, BP8, **MAN.6**, BP1, BP2, BP3, BP4, BP5,BP6, BP7, BP8, BP9, **SUP.1**, BP1, BP2, BP3, BP4, BP5, **SUP.4**, BP1, BP2,	**6.2.5,** 6.2.5.3.1, 6.2.5.3.2, **6.3.2,** 6.3.2.3.1, 6.3.2.3.2, 6.3.2.3.3, 6.3.2.3.4 **6.3.7,** 6.3.7.3.1, 6.3.7.3.2, 6.3.7.3.3	4.4 8.1, 8.2, 8.3 10.1, 10.2, 10.3,10.4,10.5	Continual Improvement • Measurement • Reporting • Improvement (The Seven-step improvement Process)

Category	SWEBOK Knowledge Areas	CMMI – DEV	CMMI Acquisition	ISO/IEC 15504	ISO/IEC 12207	PMBOK	ITIL
	Software Engineering Management • Software Engineering Measurement • Software Quality	Meas. **MA** SG1 1.1, 1.2, 1.3, 1.4, SG2, 2.1, 2.2, 2.3, 2.4 Process Improvement **OPM** SG1 SP1.1, 1.2, 1.3, SG2, SP2.1,2.2, 2.3, 2.4 SG3, SP3.1, 3.2, 3.3 **OPP** SG1 SP1.1, 1.2, 1.3, 1.4, 1.5 Reviews and Audits **PPQA** SG1 SP1.1, 1.2, SG2, SP2.1,2.2 Project **QPM** SG1 SP1.1, 1.2,1.3,1.4 SG2, SP2.1,2.2, 2.3	Meas. **MA** SG1 1.1, 1.2, 1.3, 1.4, SG2, 2.1, 2.2, 2.3, 2.4 Process Improvement **OPM** SG1 SP1.1, 1.2, 1.3, SG2, SP2.1,2.2, 2.3, 2.4 SG3, SP3.1, 3.2, 3.3 **OPP** SG1 SP1.1, 1.2, 1.3, 1.4, 1.5 Reviews and Audits **PPQA** SG1 SP1.1, 1.2, SG2, SP2.1,2.2 Project **QPM** SG1 SP1.1, 1.2,1.3, 1.4 SG2, SP2.1,2.2, 2.3	BP3, BP4, BP5,BP6, **SUP.5**, BP1, BP2, BP3, BP4, BP5, **SUP.6**, BP1, BP2, BP3, BP4, BP5,BP6,BP7, BP8, **SUP.7**, BP1, BP2, BP3, BP4, BP5,BP6, BP7, BP8,			
Organisational	None	**OT** SG1 SP1.1, 1.2, 1.3,1.4 SG2, SP2.1,2.2, 2.3	**OT** SG1 SP1.1, 1.2, 1.3,1.4 SG2, SP2.1,2.2, 2.3	**MAN.1,** BP1, BP2, BP3, BP4, BP5,BP6, BP7,**MAN.2**, BP1, BP2, BP3, BP4, BP5,BP6, BP7, **RIN.1,** BP1, BP2, BP3, BP4,	**6.2.2,** 6.2.2.3.1, 6.2.2.3.2, 6.2.2.3.3, **6.2.3,** 6.2.3.3.1, 6.2.3.3.2, 6.2.3.3.3	9.1, 9.2, 9.3, 9.4	Strategy • Demand Management • Service Strategy • IT Financial Management

Category	SWEBOK Knowledge Areas	CMMI – DEV	CMMI Acquisition	ISO/IEC 15504	ISO/IEC 12207	PMBOK	ITIL
				BP5,BP6, BP7,BP8, BP9,BP10, **RIN.2,** BP1, BP2, BP3, BP4, BP5,BP6, BP7**,RIN.3,** BP1, BP2, BP3, BP4, BP5,BP6, **RIN.4,** BP1, BP2, BP3, BP4, BP5,BP6,	**6.2.4**, 6.2.4.3.1, 6.2.4.3.2, 6.2.4.3.3, 6.2.4.3.4		• Business Relationships Management Design • Service Level Management Transition • Knowledge Management
Acquisition and Supplier Management	None	**SAM** SG1 SP1.1, 1.2, 1.3, SG2, SP2.1, 2.2, 2.3	**AM** SG1 SP1.1, 1.2, 1.3, 1.4 **SSAD** SG1 SP1.1, 1.2, 1.3, 1.4 SG2, SP2.1, 2.2, 2.3 SG3 3.1, 3.2	**ACQ.1**,BP1, BP2, BP3, BP4, BP5, BP6, **ACQ.2**, BP1, BP2, BP3, **ACQ.3**, BP1, BP2, BP3, BP4, BP5, BP6 **ACQ.4**, BP1, BP2, BP3, BP4, BP5, **ACQ.5,** BP1, BP2, BP3 **SPL.1**, BP1, BP2, BP3, BP4, BP5, BP6, BP7, BP8, BP9, BP10, **SPL.2**, BP1, BP2, BP3, BP4, BP5, BP6, BP7, BP8, BP9, BP10. BP11, **SPL.3**, BP1, BP2, BP3, BP4	6.1 Agreement Process **6.1.1,** 6.1.1.3.1, 6.1.1.3.2, 6.1.1.3.3, 6.1.1.3.4, 6.1.1.3.5, 6.1.1.3.6, 6.1.1.3.7, **6.1.2**, 6.1.2.3.2, 6.1.2.3.3, 6.1.2.3.4, 6.1.2.3.5, 6.1.2.3.6	4.6 12.1, 12.2, 12.3, 12.4	Design Phase • Supplier Management
Project Management	Software Engineering	**IPM SG1** 1.1, 1.2, 1.3, 1.4, 1.5, 1.6,	**IPM SG1** 1.1, 1.2, 1.3, 1.4, 1.5, 1.6,	**MAN.3**, BP1, BP2, BP3, BP4, BP5,BP6,	**6.3.1,** 6.3.1.3.1, 6.3.1.3.2,	4.1, 4.2, 4.3 5.1,5.2,5.3,	None

Category	SWEBOK Knowledge Areas	CMMI – DEV	CMMI Acquisition	ISO/IEC 15504	ISO/IEC 12207	PMBOK	ITIL
	Management • Initiation and Scope Definition • Software Project Planning • Software Project Enactment • Review and Evaluation • Closing	1.6 **SG2**, 2.1, 2.2, 2.3 **PMC** SG1 SP1.1, 1.2, 1.3,1,4, 1.5, 1.6, 1.7 SG2, SP2.1,2.2, 2.3 **PP** SG1 SP1.1, 1.2, 1.3,1.4 SG2, SP2.1,2.2, 2.3, 2.4, 2.5, 2.6, 2.7, SG3, SP3.1, 3.2, 3.3 **REQM** SG1 SP1.1, 1.2, 1.3, 1.4, 1.5 **RSKM** SG1 SP1.1, 1.2, 1.3 SG2, SP2.1,2.2, SG3, SP3.1, 3.2	1.6 **SG2**, 2.1, 2.2, 2.3 **PMC** SG1 SP1.1, 1.2, 1.3,1,4, 1.5, 1.6, 1.7, 1.8 SG2, SP2.1,2.2, 2.3 **PP** SG1 SP1.1, 1.2, 1.3,1.4, 1.5 SG2, SP2.1,2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8 SG3, SP3.1, 3.2, 3.3 **REQM** SG1 SP1.1, 1.2, 1.3, 1.4, 1.5 **RSKM** SG1 SP1.1, 1.2, 1.3, SG2, SP2.1,2.2, SG3, SP3.1, 3.2	BP7, BP8, BP9, BP10, BP11, BP12,BP13, BP14, BP15, **MAN.5**, BP1, BP2, BP3, BP4, BP5,BP6, BP7,	6.3.1.3.3, , **6.3.3,** 6.3.3.3.1, 6.3.3.3.2, 6.3.3.3.3, **6.3.4,** 6.3.4.3.1, 6.3.4.3.2, 6.3.4.3.3, 6.3.4.3.4, 6.3.4.3.5, 6.3.4.3.6, **6.3.6,** 6.3.6.3.1, 6.3.6.3.2	5.4,5.5 6.1,6.2, 6.3, 6.4,6.5, 6.6 7.1, 7.2, 7.3 11.1, 11.2, 11.3, 11.4, 11.5, 11.6	
Technical Engineering	Software Requirements Software Design Software Testing Software Maintenance Software Engineering Tools and Methods	**PI** SG1 SP1.1, 1.2, 1.3, SG2, SP2.1,2.2, SG3, SP3.1, 3.2, 3.3, 3.4 **RD** SG1 SP1.1, 1.2 SG2, SP2.1,2.2, 2.3 SG3, SP3.1, 3.2, 3.3, 3.4, 3.5	**ARD** SG1 SP1.1, 1.2 SG2, SP2.1,2.2 SG3, SP3.1, 3.2, 3.3, 3.4 **ATS** SG1 SP1.1, 1.2, 1.3 SG2, SP2.1, 2.2 **AVAL** SG1 SP1.1, 1.2, 1.3, SG2,	**ENG.1**, BP1, BP2, BP3, BP4, BP5, BP6, **ENG.2**, BP1, BP2, BP3, BP4, BP5, BP6 **ENG.3**, BP1, BP2, BP3, BP4, BP5, BP6, BP7,**ENG.4**, BP1, BP2, BP3, BP4, BP5, BP6, **ENG.5**, BP1, BP2, BP3, BP4, BP5	**6.4.1,** 6.4.1.3.1, 6.4.1.3.2, 6.4.1.3.3, 6.4.1.3.4, 6.4.1.3.2, **6.4.2,** 6.4.2.3.1, 6.4.2.3.3, **6.4.3,** 6.4.3.3.1, 6.4.3.3.2, **6.4.4,** **6.4.5,** 6.4.5.3.1,	None	Design • Information Security Management • Capacity Management • Availability Management • Continuity Management

Category	SWEBOK Knowledge Areas	CMMI – DEV	CMMI Acquisition	ISO/IEC 15504	ISO/IEC 12207	PMBOK	ITIL
		TS SG1 SP1.1, 1.2 SG2, SP2.1, 2.2, 2.3, 2.4 SG3, SP3.1, 3.2 **VAL** SG1 SP1.1, 1.2, 1.3, SG2, SP2.1,2.2 **VER** SG1 SP1.1, 1.2, 1.3, SG2, SP2.1,2.2, 2.3 SG3, SP3.1, 3.2	SP2.1,2.2 **AVER** SG1 SP1.1, 1.2, 1.3, SG2, SP2.1,2.2, 2.3 SG3, SP3.1, 3.2	**ENG.6**, BP1, BP2, BP3, BP4,**ENG.7**, BP1, BP2, BP3, **ENG.8**, BP1, BP2, BP3, BP4, BP5, BP6, **ENG.9**, BP1, BP2, BP3, BP4, BP5, BP6, BP7, **ENG.10**, BP1, BP2, BP3, BP4, **ENG.11**, BP1, BP2, BP3, BP4, BP5, BP6 **SUP.2**, BP1, BP2, BP3, BP4, BP5, **SUP.3**, BP1, BP2, BP3, BP4, BP5,BP6,	**6.4.6** 6.4.6.3.1, **6.4.7,** 6.4.7.3.1, **6.4.8,** 6.4.8.3.1 7.1.1 7.1.2 7.1.3 7.1.4 7.1.5 7.1.6 7.1.7		Transition • Validation and Testing • Evaluation
Support	Software Engineering Management Project Supporting Processes • Software Configuration Management • Software Release Management	**CM**. SG1,SP1.1,1.2, 1.3 SG2, SP2.1, 2.2 SG3, SP3.1, 3.2	**CM**. SG1,SP1.1,1.2, 1.3 SG2, SP2.1, 2.2 SG3, SP3.1, 3.2	**ENG.12**, BP1, BP2, BP3, BP4, BP5, BP6, **OPE.1**, BP1, BP2, BP3, BP4, BP5, **OPE.2**, BP1, BP2, BP3, BP4, BP5, BP6, **SUP.8**, BP1, BP2, BP3, BP4, BP5,BP6, BP7, BP8, PB9, PB10, **SUP.9**, BP1, BP2, BP3, BP4, BP5,BP6, BP7, BP8, PB9, PB10, **SUP.10**, BP1, BP2,	**6.3.5,** 6.3.5.3.1, 6.3.5.3.2, **6.4.9,** 6.4.9.3.1, 6.4.9.3.2, 6.4.9.3.3, 6.4.9.3.4, 6.4.9.3.5 **6.4.10,** 6.4.10.3.1, 6.4.10.3.2, 6.4.10.3.3, 6.4.10.3.4, 6.4.10.3.5,	4.5 5.4, 5.5	Transition • Change Management • Configuration Management • Release and Deployment Management Service Operation • Incident Management

Category	SWEBOK Knowledge Areas	CMMI – DEV	CMMI Acquisition	ISO/IEC 15504	ISO/IEC 12207	PMBOK	ITIL
	and Delivery			BP3, BP4, BP5,BP6, BP7, BP8, PB9	**6.4.11,** 6.4.11.3.1, 6.4.11.3.2,		• Request Fulfilment • Problem Management

Appendix C - Delphi Study Instruments and Tools

11.1 Invitation Letter Posted on LinkedIn Group Forum

Dear Sir/Madam, Date: 5 April 2014

I would like to invite you to become a Delphi panel member to evaluate a software development process capability framework developed as part of my Masters study at Auckland University of Technology – New Zealand. The research attempts to gain expert consensus on a set of software capabilities and underlying processes that compromise a proposed framework. With the help of experts I attempt to evaluate the framework to make it feasible for application in small software enterprises. The framework is derived from the following standards:

1. A Guide to the Project Management Body of Knowledge (PMBOK)
2. Capability Maturity Model Integration for Development (CMMI-DEV V 1.3)
3. Software Process Improvement and Capability Determination (ISO/IEC 15504)
4. Software life cycle processes (ISO/IEC 12207, 2008)
5. Information Technology Infrastructure Library (ITIL V3)

Your long expertise (minimum 5 years) qualifies you to become a panel member. Participating members can be process improvement and quality assurance consultants, project managers, software solution architects, system analysts and personnel who have direct involvement in the software development projects and any other relevant experience. A number of panel members must have exposure to small software organisations.

The study will follow a Delphi format which involves 3 rounds survey combined with controlled feedback. Each survey will only take 20 – 40 minutes. The research will be conducted in accordance with Auckland University of Technology ethics and governance regulations. As a panel member you can receive feedback on the outcomes of this research.

For further information and to express an interest in participating in this project, please contact me on xxxxx@xxxxxl.com . Once you express your interest, I will forward to you the Participant Information Sheet to explain your role as panel member.

Please forward this message to other professionals or colleagues who may be interested in taking part.

Yours Sincerely

Fatina Aweidah

11.2 Consent Form

Consent Form

AUT UNIVERSITY
TE WĀNANGA ARONUI O TAMAKI MAKAU RAU

Project title: Holistic Approach to Software Process Improvement in Small Software Enterprises: Delphi Study

Project Supervisor: ***Prof Felix Tan***
Researcher: ***Fatina Aweidah***

☐ I have read the Participant Information Sheet and I understand the nature of the research, why I have been selected and what is expected of me.

☐ I have had the opportunity to ask questions and have them answered to my satisfaction.

☐ I agree to take part in this research.

☐ I understand that I am free to stop participating at any time, and to withdraw any data traceable to me up to three weeks after the last interview.

☐ I understand that the information collected from the questionnaire will be aggregated and anonymized, so that it will not be linked to any particular individual(s).

☐ I understand that I will not be identified in any publications resulting from this research

☐ I understand that data will be kept for at least 6 years, and will be destroyed at the end of the research period

Participant's Signature: ....

Participant's name: ....

Participant's Contact Details (if appropriate): ……………………………………………………….

………..

Date:

Approved by the Auckland University of Technology Ethics Committee on 5 September 2013 AUTEC Reference number 13/221

Note: The Participant should retain a copy of this form.

11.3 Participant Information Sheet

Participant Information Sheet

Date Information Sheet Produced

13 April 2014

Project Title

Holistic Approach to Software Process Improvement in Small Software Enterprises: A Delphi Study

An Invitation

I am a student at the department of Business Information Systems at Auckland University of Technology conducting this research as part of my Masters study. I would like to invite you to participate in this research project. Once you accept participation in this study by replying to my email, I will forward to you the ***Survey Questionnaire*** to complete and supporting ***Project Information Sheet*** which provides more details about the research context, goals, and about the proposed capability model and related processes.

What is the purpose of this research?

This research study intends to propose a framework that will guide small software development enterprises (SSEs) in their process improvement initiatives by helping them adopt number of processes in a number of areas that would ensure consistent delivery of software development projects across the organisation.

How was I identified and why am I being invited to participate in this research?

You have been selected as a potential participant because of your research, work experience and knowledge in software process improvement programs for software development organisations. Your name and contact details have been obtained from colleagues in the industry, from linked in social forum or, from publications in this field and/or after contacting me expressing your willingness to participate in this study. Your participation makes you a member of an expert panel who will have a say on a model of organisational practices developed proposed for small software enterprises (SSE).

What will happen in this research?

Participation in this research project will involve taking part in a Delphi Study. A Delphi study is a structured technique for collecting expert opinion on an area of research by means of survey questionnaire. The process can be iterated in a number of rounds until a consensus have been achieved.

It is anticipated that the Delphi study will be conducted in three rounds. In the first round the survey questionnaire will be distributed. Panellists will be asked to review the proposed model and write comments and recommendations on the suggested processes for each capability. In the second round, the collective responses and recommendations of respondents will be summarised and presented to panellists. Panellists will be given the chance to change their opinion/ recommendations. The survey will include more structured questions in round 2. In round 3 the survey will be very similar to round 2, however panellists will be asked to reach a consensus on the recommended processes for each capability.

What are the discomforts and risks?

Participation is completely voluntary and it is anticipated that there this study not cause any risks or discomforts to participants in this study. Participants will remain confidential through all stages of the study.

How will these discomforts and risks be alleviated?

This research will not cause any discomforts and risks for participants. The researcher will not identify any individuals or attribute information or comments to any organisation. Your name and the name of organisation you represent will not be exposed to anyone. The researcher will only have access to the project data. The summary of the research results may be used in book chapters, journals and conference publications without any identifying information.

What are the benefits?

Small Software enterprises (SSEs) comprise the largest population of the world software industry (Laporte, Alain, & Renault, 2006; Kirk & MacDonell, 2009; Carter-Steel A., 2001). The software development community is showing and increasing interest in SPI initiatives in SSEs (Pino, Pedreira, Garcia, Rodriguez, & Piattini, 2010). This study proposes to identify a comprehensive set of capabilities that will help SSEs to obtain the capacity to develop sound processes that underpin the quality of the software product and software projects through all stages of the product lifecycle. Details of the proposed model will be emailed to you with the survey once you consent your participation in this study.

How will my privacy be protected?

All data will be held in secure locations within AUT University in compliance with Auckland University of Technology Ethics Committee (AUTEC) regulations.

What are the costs of participating in this research?

The time commitment will be 20-40 minutes. Your participation is integral for developing a reference process model endorsed by experts in this field that can be used in small software enterprises. Should you prefer not to participate, or you wish to cease participation at any time, you are free to do so.

What opportunity do I have to consider this invitation?

One week.

How do I agree to participate in this research?

By signing and sending back the attached consent form.

Will I receive feedback on the results of this research?

If you wish to receive feedback on the outcomes of this research, you may contact me by email.

What do I do if I have concerns about this research?

Any concerns regarding the nature of this project should be notified in the first instance to the Project Supervisor, Prof. Felix B. Tan, Faculty of Business and Law, AUT University, Private Bag 92006, Auckland 1142, Email: felix.tan@aut.ac.nz, Phone: +649 9219487

Concerns regarding the conduct of the research should be notified to the Executive Secretary, AUTEC,
Kate O'Connor, Private Bag 92006, Auckland 1010, Email ethics@aut.ac.nz, Phone +64 9 921 9999 extn: 6038

Whom do I contact for further information about this research?

Researcher Contact Details:

Fatina Aweidah,
Masters Student at the Faculty of Business and Law, Auckland University of Technology (AUT)
Email: xxxx@xxxx.com
Phone numbee: xxxxx

Project Supervisor Contact Details:

Prof. Felix B. Tan, Faculty of Business and Law, AUT University, Private Bag 92006, Auckland 1142, Email: xxxxxxxxx, Phone: xxxxxxxx

Ethics Approval

Approved by the Auckland University Of Technology ethics committee on 5 September 2013 AUTEC Reference number 13/221

11.4 First Round Questionnaire

Software Process Capability Model for Small Software Enterprises

Round-1 Questionnaire

Thank you very much for participating in this Delphi study. This is the first Delphi round; It is expected that three Delphi rounds will be required to complete the study. The number of rounds required will be dependent on reaching a consensus at the end of each round.

In this survey your role is to help the researcher to identify the necessary processes for ensuring successful projects. This study structures the processes/practices under five identified core capabilities:

1. Process Definition and Improvement
2. Organisational
3. Project Management
4. Technical Solution Development
5. Support

Your role as a panel member is to:

- evaluate the processes proposed by the researcher for each capability
- add or remove processes as you deem appropriate. Please be aware that the suggested processes were based on practices suggested by existing standards used in the study [Refer to information sheet].
- identify dynamic capabilities that can help the organisation to survive in highly dynamic markets.
- evaluate the feasibility of the practices in the context of ***small software enterprises*** (SSE).

It is hoped that the outcome of this study is a process model workable for SSE. The model should include the capabilities and processes that would help SSE gain control over their software development and maintenance projects.

Respondents to this study will remain anonymous to each other but not to the researcher. For this purpose, respondents will be required to provide an email address on the questionnaire. The e-mail addresses and all identifying information will be kept strictly confidential. Your participation in this study is voluntary.

A summary of the final results of this study will be made available to you using the e-mail addresses supplied. The study will further be used for a Master's thesis. The details of the supervisor for this study: *Dr Felix B. Tan, Professor of Information Systems and Head of Department, Faculty of Business and Law, Auckland University of Technology*

Please complete the survey and email back to me on email: xxxxxxxxxxxxxxxx. ***Please ensure to use the attached Project Information Sheet to be able to answer the survey questions.***

Following are the definitions of the scales:

(4) Very Highly Important	Highest relevance to the model Highest impact on the organisation, first priority.	(4) Very Highly Feasible	Can definitely be implemented No increase in available resources needed. Acceptable to all in the organisation.
(3) Very Important	High relevance to the model, High measurable impact on organisation, second priority.	(3) Highly Feasible	Can be implemented No major increase in available resources needed. Acceptable to majority in the organisation.
(2) Important	Relevant to the model Has a measurable impact on the organisation, low priority.	(2) Feasible	Can be implemented. Slight increase in available resources needed. Acceptable to majority in the organisation.
(1) low importance	Very low relevance to the model Very low measurable impact on the organisation, no priority.	(1) Not Very Feasible	Some indication that it cannot be implemented. Major increase in available resources needed. Unacceptable to majority in the organisation.
(0) not needed, remove	The practice has no relevance and can be removed from the model.	(0)Not Feasible (Remove)	Cannot be implemented, Extremely large scale increase in available resources needed, Completely unacceptable to all in the organisation.

Part I - Demographic Information

The following demographic information will be used for purposes of data analysis. The information will not be used for any other purpose. The privacy of all respondents is guaranteed.

1. Please write in the space below your email address:

2. What geographical region are you from?

- Europe
- Australia and New Zealand
- Sourth Asia
- West Asia
- Africa
- South America
- North America

3. What is the number of employees in your organisation?

- 1-10
- 10- 30
- 30- 50
- More than 50

4. How many years of experience do you have?

5. Do you have experience in one or more of the following roles?

- Software Process Improvement
- Software Process Measurements
- Software Quality Assrunace
- Software Organistions Audits
- Project Management
- System Analyst
- Solutions Architect
- Software Development
- Quality Assurance
- Software Testing
- Release & Deployment
- Configuration Manager
- Software Maintenance
- Other (Please Specify)

6. What is your highest qualification? (Optional)

- Undergraduate Degree
- Post Graduate Degree

- Graduate Diploma
- Other relevant certifications or qualifications (Please Specify)

7. **What is the monetary value of software development related projects you are involved in?**
   - Less than $50,000
   - 50,000 - $100,000
   - $100,000 - $500,000
   - $500,000 - $1000,000
   - More than $1000,000

8. **Have you had experience in the context of small software enterprises (SSEs)?** (*50 employees or less*)

Part II - Study Related Questions

1. **The research study has identified 5 capabilities/process groups for small software enterprises. Please rate the capability by using the ratings below:**

Importance:	(0) not needed (Remove) (1) low importance (2) important (3) highly important (4) very highly important	***Feasibility*** *for SSE*	(0)not Feasible (Remove) (1) not very feasible (2) feasible (3) highly feasible (4) very highly feasible

Capability/Process Group	Importance (0-4)	Feasibility for SSE (0-4)	Comments
1.Process Definition and Improvement			
2.Organisational			
3.Technical Solution Development			
4.Project Management			
5.Support			

2. Do you believe the Dynamic Capabilities (DCs) approach is feasible in the context of organisations involved in the development of software or does it add an unneeded level of complexity?

3. Do you think the Dynamic Capabilities (DCs) theory is feasible in the context of small software enterprises or does it add an unneeded level of complexity?

4. Are you in favour of specific international standards for small software organisations over other standards? Please explain your reasons for taking this position.

Part III. Evaluation of Process Definition & Improvement capability

Please evaluate the suggested processes for the proposed process capability model by using the ratings below:

Importance *for SSE:*	(0) not needed (Remove) (1) low importance (2) important (3) highly important (4) very highly important	***Feasibility*** *for SSE:*	(0)not Feasible(Remove) (1) not very feasible (2) feasible (3) highly feasible (4) very highly feasible

1.PROCESS DEFINITION AND IMPROVEMENT			
1.1 PROCESS ESTABLISHMENT **(DC Enabler)**	Importance (0-4)	Feasibility (0-4)	**What is your motivation for your ratings of importance and feasibility?** ***[Please provide comments to help the researcher compare your reasons with other panel members]***
1.1.1 Initiate process establishment			
1.1.2 Define a process			
1.1.3 Identify work products			
1.1.4 Define architecture			
1.1.5 Define control mechanism			
1.1.6 Define process measurements			
1.1.7 Document process			

1.2 PROCESS REVIEW	Importance (0-4)	Feasibility for (0-4)	What is your motivation for your ratings of importance and feasibility? *[Please provide comments to help the researcher compare your reasons with other panel members]*
1.2.1 Plan process review			
1.2.2 Collect and analyse process data			
1.2.3 Review process performance (DC Sense)			
1.2.4 Document review result (DC Seize)			
1.2.5 Report review results			

1.3 PROCESS IMPROVEMENT	Process Importance (0-4)	Process Feasibility (0-4)	What is your motivation for your rating of the process importance and feasibility? *[Please provide comments to help the researcher compare your reasons with other panel members]*
1.3.1 Identify and prioritise process improvement (DC Sense, Seize)			
1.3.2 Plan improvements (DC Seize)			
1.3.3 Implement process action plans (DC Transform)			
1.3.4 Monitor and evaluate improvement (DC Sense, Seize)			
1.3.5 Update organisation process asset library			

1. In general how do you evaluate the above capability?

2. Would you recommend any additional practices to the above?

3. Do you agree with the suggested dynamic capabilities (DCs)?

4. Would you suggest any additional DCs for this Capability?

5. Any other comments you would like to add?

*[Next, please evaluate **Organisational** capability]*

Part IV - *Organisational capability*

Please evaluate the suggested processes for ***ORG*** *capability by using the ratings below:*

Importance:	(0) not needed (Remove) (1) low importance (2) important (3) highly important (4) very highly important	***Feasibility*** *for SSE*	(0)not Feasible(Remove) (1) not very feasible (2) feasible (3) highly feasible (4) very highly feasible

2. ORGANISATIONAL			
2.1 BUSINESS ALIGNMENT	Importance (0-4)	Feasibility (0-4)	**What is your motivation for your rating of importance and feasibility?** ***[Please provide comments to help the researcher compare your reasons with other panel members]***
2.1.1 Identify vision and goals and develop business objectives (DC Sense)			
2.1.2 Identify business outcomes to achieve strategic objectives (DC Seize)			
2.1.3 Achieve strategic IT plans. (DC Transform)			

2.2 CUSTOMER RELATIONSHIP	Importance (0-4)	Feasibility for SSE (0-4)	**What is your motivation for your rating of importance and feasibility?** ***[Please provide comments to help the researcher compare your reasons with other panel members]***

2.2.1 Maintain Customer Relationships (DC Sense)			
2.2.2 Sign up Contracts (DC Seize, Transform)			
2.2.3 Customer Satisfaction Survey (DC Sense)			
2.2.4 Analyse Customers' surveys and develop action plans. (DC Seize and Transform)			

2.3 FINANCIAL PLANNING	Importance (0-4)	Feasibility for SSE (0-4)	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
2.3.1 Develop IT Budget			
2.3.2 Financial Analysis and reporting			
2.3.3 Invoicing			

2.4 HUMAN RESOURCE PLANNING	Importance (0-4)	Feasibility for SSE (0-4)	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
2.4.1 Identify needed skills and competencies **(DC Sense)**			
2.4.2 Recruit qualified staff **(DC Seize, Transform)**			

2.4.3 Evaluate and develop staff skills			
2.4.4 Plan training needs **(DC Seize)**			
2.4.5 Deliver and maintain staff training (DC Transform)			

2.5 KNOWLEDGE MANAGEMENT	**Importance (0-4)**	**Feasibility for SSE (0-4)**	**What is your motivation for your rating of importance and feasibility?** ***[Please provide comments to help the researcher compare your reasons with other panel members]***
2.5.1 Develop KM infrastructure			
2.5.2 Establish a network of experts (DC Sense)			
2.5.3 Capture and disseminate knowledge (DC Seize, transform)			
2.5.4 Maintain knowledge system (DC Sense, Seize)			

1. In general how do you evaluate the above capability?

2. Would you recommend any additional practices to the above?

3. Do you agree with the suggested dynamic capabilities (DCs)?

4. Would you suggest any additional DCs for this Capability?

5. Any other comments you would like to add?

[Next, please evaluate Technical Solution Development capability]

Part V - *Technical Solution Development* capability

Please evaluate the suggested processes for the proposed process capability model by using the ratings below:

Importance:	(0) not needed (Remove) (1) low importance (2) important (3) highly important (4) very highly important	***Feasibility*** *for SSE*	(0)not Feasible(Remove) (1) not very feasible (2) feasible (3) highly feasible (4) very highly feasible

3. TECHNICAL SOLUTION DEVELOPMENT			
3.1 SOFTWARE REQUIREMENTS	Importance (0-4)	Feasibility (0-4)	**What is your motivation for your rating of importance and feasibility?** ***[Please provide comments to help the researcher compare your reasons with other panel members]***
3.1.1 Obtain software scope			
3.1.2 Elicit requirements			
3.1.3 Analyse requirements			
3.1.4 Trace requirements			
3.1.5 Validate and prioritise requirements			
3.1.6 Develop requirements baseline			
3.1.7 Develop test cases			

3.2 SOFTWARE DESIGN	Importance (0-4)	Feasibility (0-4)	What is your motivation for your rating of importance and feasibility? *[Please provide comments to help the researcher compare your reasons with other panel members]*
3.2.1 Develop software architecture			
3.2.2 Verify design			
3.2.3 Update test cases			

3.3 SOFTWARE CONSTRUCTION	Importance (0-4)	Feasibility (0-4)	What is your motivation for your rating of importance and feasibility? *[Please provide comments to help the researcher compare your reasons with other panel members]*
3.3.1 Develop coding standards			
3.3.2 Obtain complete test cases			
3.3.3 Code software products			
3.3.4 Verify individual software units			
3.3.5 Verify the integrated units			

3.4 SOFTWARE TESTING	Importance (0-4)	Feasibility (0-4)	What is your motivation for your rating of importance and feasibility? *[Please provide comments to help the researcher compare your reasons with other panel members]*
3.4.1 Apply system performance testing			
3.3.2 Validate the system			
3.3.3 Perform regression testing *			

In general how do you evaluate the above capability?

1. Would you recommend any additional practices to the above?

2. Do you agree with the suggested dynamic capabilities (DCs)?

3. Would you suggest any additional DCs for this Capability?

4. Any other comments you would like to add?

[Next, please evaluate Project Management capability]

Part VI - *Project Management* capability

Please evaluate the suggested processes for the proposed process capability model by using the ratings below:

Importance:	(0) not needed practice (Remove) (1) low importance (2) important practice (3) highly important practice (4) extremely important practice	***Feasibility*** *for small project settings*	(0)not Feasible(Remove) (1) not very feasible (2) feasible (3) highly feasible (4) very highly feasible

4. PROJECT MANAGEMENT			
4.1 PROJECT INITIATION	**Importance (0-4)**	**Feasibility (0-4)**	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
4.1.1 Perform feasibility analysis (DC Sense, Seize)			
4.1.2 Initiate scope			
4.1.3 Initiate schedule			
Initiate budget			
4.1.4 Initiate communication management			
4.1.5 Initiate Team			
4.1.6 Initiate risk analysis (DC Sense)			
4.1.7 Initiate supplier management (DC Sense, Seize, Transform)			

4.2 PROJECT PLANNING	Importance (0-4)	Feasibility (0-4)	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
4.2.1 Plan scope			
4.2.2 Plan schedule			
4.2.3 Plan cost			
4.2.4 Plan quality			
4.2.5 Plan communication			
4.2.6 Plan human resources			
4.2.7 Plan risks (DC Seize)			
4.2.8 Plan supplier management (DC Seize)			

4.3 PROJECT EXECUTION	Importance (0-4)	Feasibility (0-4)	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
4.3.1 Execute scope			
4.3.2 Execute schedule			
4.3.3 Execute cost			
4.3.4 Execute quality management plan (DC Sense, Seize, Transform)			

4.3.5 Perform communication management (DC Sense, Seize, Transform)			
4.3.6 Execute human resource management			
4.3.7 Execute risk management plan			
4.3.8 Perform supplier management (DC Sense, Seize, Transform)			

4.4 PROJECT MONITORING & CONTROLLING (DC Sense, Seize)	Importance (0-4)	Feasibility (0-4)	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
4.4.1 Integrated Change Control			
4.4.2 Monitor and control scope			
4.4.3 Monitor and control schedule and cost			
4.4.4 Monitor and control cost			
4.4.5 Monitor and control quality			
4.4.6 Monitor and control communication			
4.4.7 Monitor and control human resources			
4.4.8 Monitor and control risks			
4.4.9 Monitor and control supplier			

5.PROJECT CLOSING	Importance (0-4)	Feasibility (0-4)	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
5.1 Close Supplier Contract			
5.2 Update Project Documentation (DC Sense, Seize)			

1. In general how do you evaluate the above capability?

2. Would you recommend any additional practices to the above?

3. Do you agree with the suggested dynamic capabilities (DCs)?

4. Would you suggest any additional DCs for this Capability?

5. Any other comments you would like to add?

[Next, please evaluate Support Capability]

Part VII - Evaluation of ***Support*** Capability

Please evaluate the suggested processes for the proposed process capability model by using the below ratings:

Importance*:*	(0) not needed practice (Remove) (1) low importance (2) important practice (3) highly important practice (4) very highly important practice	***Feasibility*** *for small project settings*	(0)not Feasible(Remove) (1) not very feasible (2) feasible (3) highly feasible (4) very highly feasible

5.Service Support Capability			
5.1 SERVICE LEVEL MANAGEMENT	**Importance (0-4)**	**Feasibility (0-4)**	**What is your motivation for your rating of importance and feasibility?** ***[Please provide comments to help the researcher compare your reasons with other panel members]***
5.1.1 Identification of service requirements			
5.1.2 Agreements sign-off			
5.1.3 Service level monitoring and reporting			

5.2 CHANGE MANAGEMENT	**Importance (0-4)**	**Feasibility (0-4)**	**What is your motivation for your rating of importance and feasibility?** ***[Please provide comments to help the researcher compare your reasons with other panel members]***
5.2.1 Review of request for change			
5.2.2 Assessment of change request			

5.2.3 Change evaluation and authorisation			
5.2.4 Change Deployment Authorisation			
5.2.5 Post Implementation Review and Change Closure (DC Sense, Seize)			

5.3 RELEASE MANAGEMENT	**Importance (0-4)**	**Feasibility (0-4)**	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
5.3.1 Release Planning			
5.3.2 Establish the release infrastructure			
5.3.3 Release documentation			
5.3.4 Deploy the release			

5.4 CONFIGURATION MANAGEMENT (DC-Enabler)	**Importance (0-4)**	**Feasibility (0-4)**	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
5.4.1 Identify Configuration Items			
5.4.2 Establish a Configuration Management System			

5.4.3 Control Configuration Items			
5.4.3 Perform Configuration Audits			

5.5 PROBLEM MANAGEMENT	**Importance (0-4)**	**Feasibility (0-4)**	**What is your motivation for your rating of importance and feasibility?** *[Please provide comments to help the researcher compare your reasons with other panel members]*
5.5.1 Proactive Problem Identification			
5.5.2 Problem Categorization and Prioritization			
5.5.3 Problem Diagnosis and Resolution			
5.5.4 Problem Closure and Evaluation (DC Seize)			
5.5.5 Major Problem Review (DC Sense)			
5.5.6 Problem Management Reporting			

1. In general how do you evaluate the above capability?

2. Would you recommend any additional practices to the above?

3. Do you agree with the suggested dynamic capabilities (DCs)?

4. Would you suggest any additional DCs for this Capability?

5. Any other comments you would like to add?

[End of Survey]

11.5 First Round Information Sheet

Project Information Sheet for Participants

Project Title: Holistic Approach to Software Process Improvement in Small Software Enterprises: A Delphi Study

Student Name: Fatina Aweidah
(Masters student, Auckland University of Technology, Faculty of Business and Law, Department of Business Information Systems)

Research Overview

The establishment of organisational processes to improve and standardise the development of software products and services is paramount to the success of software development organisations. Although mid to large-size organisations have begun to comprehend the importance and the benefits of adopting international standards, there is a strong belief among researchers and practitioners that existing standards are not suitable for small-size enterprises (SSEs).

The software industry in most countries is mainly composed of small enterprises, which also contribute to the development of national economies. According to the New Zealand Ministry of Development (2010), 97.2% of enterprises in New Zealand employ 19 or fewer employees, and SMEs dominate most industries.

This study targets small software organisations that are involved in the development of the software and the aspects of services needed to maintain and support the product after being released. In terms of organisation size the focus will be on small enterprises (SEs) with 25 -50 employees or fewer as well as IT departments in medium and large size enterprises with very small structure.

In an effort to adopt a more holistic approach, many organisations are adopting multiple frameworks to accommodate areas not addressed by a single international standard. The different SPI standards focus on a specific part of the business. For example, PMBOK focuses on project management practices, ITIL focuses on service management practices, ISO/IEC 12207 focus is on technical engineering. This research study is proposing a comprehensive software process model which is expected to help small size organisations to focus on the project-specific objectives as well as business priorities and will help them meet the requirements of multiple standards. The model is informed by the established practices adopted by the following international standards:

- Software Engineering Body of Knowledge (SWEBOK)
- A Guide to the Project Management Body of Knowledge (PMBOK)
- Capability Maturity Model Integration for Development (CMMI-DEV V 1.3)
- Software Process Improvement and Capability Determination (ISO/IEC 15504)
- Software Life Cycle Processes (ISO/IEC 12207, 2008)
- Information Technology Infrastructure Library (ITIL V3)

A critical driver underlying the success of software organisations undertaking software development is the development of a number of technical and organisational capabilities. This research study has identified five core capabilities, which small software organisations need to develop to reach high level of maturity in managing successful and consistent software development projects. In identifying the capabilities, the study draws on earlier research and the analysis of existing standards. The following section provides a description and suggests the practices needed for each of the five identified process areas.

6. Process Definition and Improvement
7. Organisational
8. Project Management
9. Technical Solution Development
10. Support

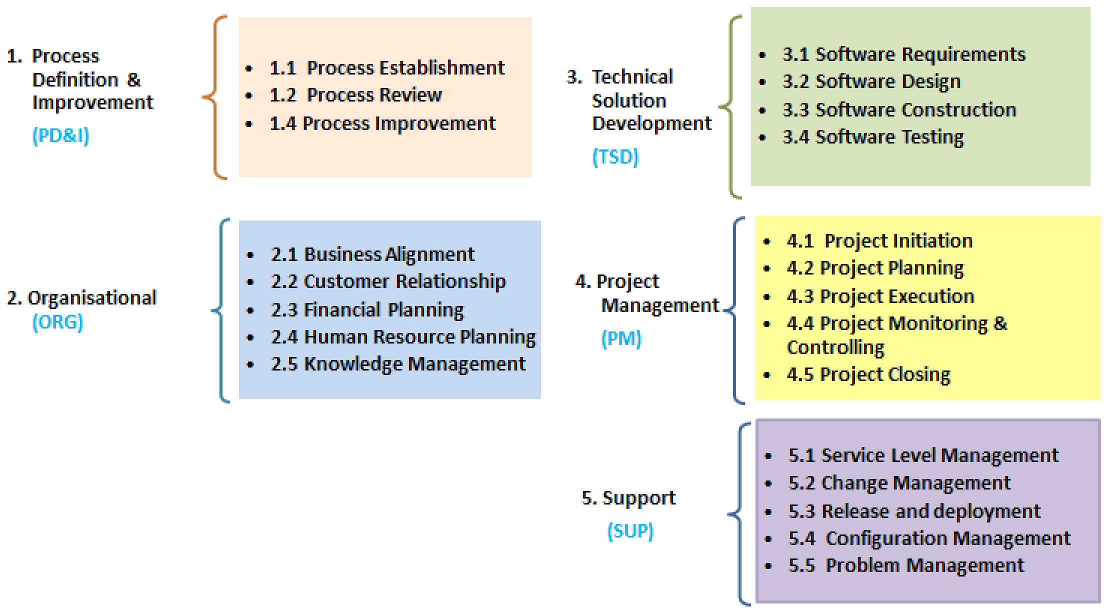


Figure 1: The proposed Software Process Capability Model

Research is showing that modern business enterprises are operating in a highly competitive or fast changing business environment, referring to major drastic and unforeseen changes in technological, competitive and regulatory domains. In such markets firms are in need of effective capabilities known as "Dynamic Capabilities". These capabilities will enable the firm to identify external opportunities and changing customer needs as they happen, as well a to identify internal opportunities for improvement, then react to these challenges fast and appropriately before the strategic significance is lost.

Therefore, managers of software development organisations require processes that will help them ***sense***, ***seize*** and ***transform*** opportunities and create new sources of competitive advantage.

Figure 2: DC Model

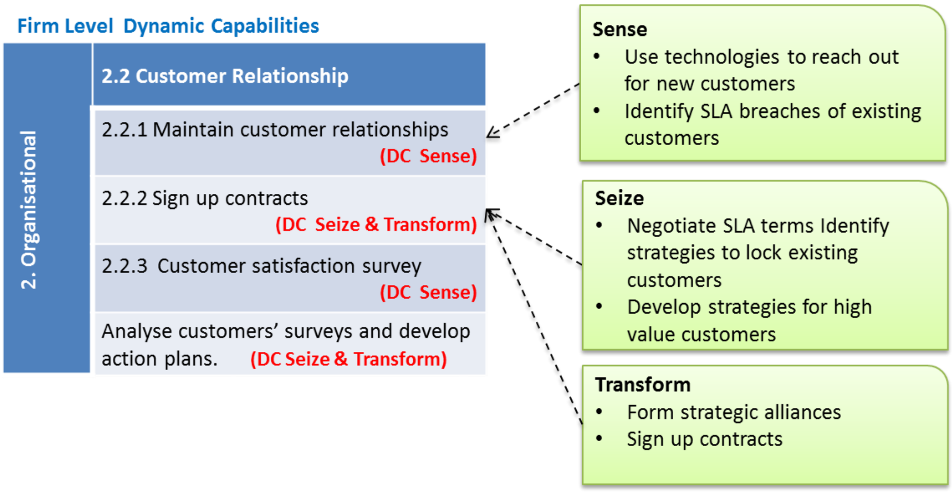
They should be able to make the best assessment about the external environment (customers, competitors and suppliers) and the internal environment (strengths, weaknesses and the availability of tangible and intangible resources) and ensure a match between them.

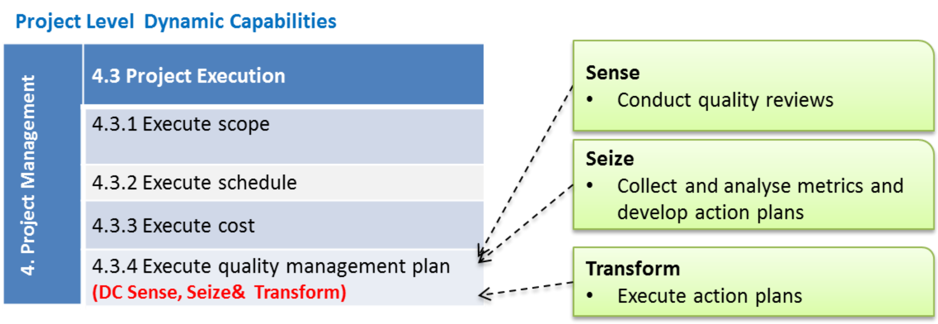
Adopting a sense and respond approach can help small firms develop external relationships to compensate for their limited resources. The literature reports that software organisations tend to limit the adoption of sense and respond practices at the project level to help managers respond to customer needs, overlooking the need to bring them up to an organisation level (Schmidt & Mathiassen, 2009). The firm's adaptability can be increased by combining project and firm level capabilities (Schmidt & Mathiassen, 2009).

This view has led the researcher to identify a number of practices in the proposed model as sense, seize and transform dynamic capabilities.

Following are examples of Dynamic Capabilities:

- Process focused on customer and supplier relations.
- Processes focused on strategic alliance.
- Processes that evaluate market and competitors.
- Processes linking customer experience with engineering design choices.
- Processes for gathering and processing internal and external information.
- Processes that allow inter-organisational learning.
- Entrepreneurial practices





Research Goal

The ultimate aim of this study is to propose a process reference model that will guide software development organisations in their process improvement initiatives. This proposed model will seek to:

1. provide the needed processes to help organisations meet the perceived project outcomes.
2. identify dynamic capabilities that will help organisations survive in highly dynamic environments.
3. build upon existing software engineering standards.
4. provide a multi-faceted approach without the need to implement multiple SPI standards.
5. be feasible for application in small software organisations.

Research Method and Role of Panellist

The study will employ the Delphi method to evaluate the usefulness of the proposed model for small software organisations in 3 rounds. Panellists will be asked to recommend changes to the model (remove, add capabilities and processes) to make it workable for SSEs

Participation and Participants Rights

Participation is completely voluntary and it is anticipated that there will not be any risk to participants in this study. Participants will remain anonymous through all stages of the study; their responses and personal data will be kept in a locked filing cabinet at AUT for six years after completion of the study; only the researcher will have access to the data. The summary of the research results may be used in book chapters, journals and conference publications without any identifying information. If you decide to participate, you have the right to withdraw from the study at any stage, although your contribution to the study before withdrawal will be used unless you instruct the researcher not to do so. All your responses will then be destroyed.

Delphi Rounds

It is anticipated that the Delphi study will be conducted in three rounds. In the first round the survey questionnaire will be distributed. Panellists will be asked to review the proposed model and write comments and recommendations on the suggested processes for each capability. In the second round, the collective responses and recommendations of respondents will be summarised and presented to panellists. Panellists will be given the chance to change their opinion/recommendations. The survey will include more structured questions in round 2. The survey in round 3 is very similar to round 2; however, panellists will be asked to reach a consensus on the recommended processes for each capability.

The result of the Delphi study will be communicated to all members of the panel. During the three rounds, communication will occur via email. Respondents' confidentiality will be guaranteed.

Time Requirements

Each round of the Delphi Study should take participants between 20 to 40 minutes to complete, depending on the depth of their answers. Following is the planned schedule for conducting the Delphi Study:

Round	Commencement Date	Survey Submission Date before
Delphi - Round 1		
Questionnaire sent to panel members	Monday, 5 May 2014	Sunday, 25 May 2014
Analysis of results and development feedback sheet	Monday, 26 May 2014	Sunday, 15 June 2014
Delphi - Round 2		
Questionnaire sent to panel members	Monday, 16 June 2014	Sunday,6 July 2014
Analysis of results and development feedback sheet	Monday, 7 July 2014	Sunday, 27 July 2014
Delphi - Round 3 (if Needed)	TBD	TBD
Summary of Findings These will be provided to participants who wish to receive them.	TBD	TBD

Ethics Approval

Ethics permission to conduct this study has been obtained. Should you have any complaints concerning the manner in which this research is conducted, please contact the supervisor.

Capabilities and Related Processes

In the following pages is a description of the five capabilities and their related processes. Please read this carefully and use it in conjunction with the survey questionnaire.

1. PROCESS DEFINITION AND IMPROVEMENT

This capability will enable the organisations to: a) institute processes in the enterprise through the development of policies and procedures and employee training programs. b) develop and sustain measurements that will enable them to gain insight into their performance and assess the effectiveness of their processes. c) provide continuous improvement to their processes.

1.1. PROCESS ESTABLISHMENT (DC Enabler)

1.1.1. Initiate process establishment	• Assign a group to facilitate and manage process establishment. Assign budget and resources.
1.1.2. Define a process	• Define and maintain a description of each process and main business objectives. This shall include the process main purpose, deliverables expected and acceptance criteria.
1.1.3. Identify work products	• Identify the outputs of the selected process. This includes detailed tasks and activities to carry out a process.
1.1.4. Define architecture	• Identify all the relationships (interfaces) with other processes or sub-processes. • Identify the target processes that will receive the WP as input. • Identify the requirements to be catered for.
1.1.5. Define control mechanism	• Identify roles and responsibilities
1.1.6. Define process measurements	• Define a set of process measures. • Specify procedures for collecting, updating and retrieving measures.
1.1.7. Document process	• Develop documents guiding the execution of the process. This may be a single procedure or a set of procedures, guidelines, work instructions and whatever else is needed.

1.2. PROCESS REVIEW

1.2.1. Plan process review	• Develop process assessment plan. Assign budget, timeline and resources.
1.2.2. Collect and analyse process data	• Carry out the review to collect process data and measurements.
1.2.3. Review process performance (DC Sense, Seize)	• Analyse process data to understand relative strengths and weaknesses of the process.
1.2.4. Document review results	• Maintain the results of the review using a standard format in an accessible location.
1.2.5. Report review results	• Communicate the results of the reviews and the action plans needed to the stakeholders.

1.3. PROCESS IMPROVEMENT

1.3.1. Identify and prioritise process Improvements (DC Sense, Seize)	• Identify organisation objectives and prioritise improvements. Organisation commits to the improvement
1.3.2. Plan Improvements (DC Seize)	• Develop process Improvement plan. • Assign budget, timeline, resources; identify scope, and develop improvement action plans.
1.3.3. Implement process action plans (DC Transform)	• Test Improvement action plans identified in the review process on a small group. • Perform full implementation across the organisation.
1.3.4. Monitor and evaluate improvements (DC Seize)	• Monitor process implementation and verify against improvement goals.
1.3.5. Update organisation process asset library (Dc Sense, Seize)	• Derive lessons learned from defining and implementing organizational process assets. • Make lessons learned available to relevant parties.

[Next Organisational Capability]

2. ORGANISATIONAL CAPABILITY

The main purpose of this capability is to introduce processes to support the strategic vision and direction of the business and to achieve alignment with the IT and software development functions in the organisation leading to the development of successful products and services. The literature review stressed certain characteristics of SSE such as the lack of strategic planning, high employee turnover and the absence of knowledge management practices.

2.1. BUSINESS ALIGNMENT

2.1.1. Identify vision and goals and develop business objectives (DC Sense)	• Evaluate changing economic conditions of existing market spaces. • Identify new market spaces, new customers and business opportunities to achieve strategic objectives.
2.1.2. Identify business outcomes to achieve strategic objectives (DC Seize)	• Identify desired business outcomes, assign responsibilities and authorise new investments to achieve business objectives
2.1.3. Achieve strategic IT plans. (DC Transform)	• Plan and implement business outcomes; assess whether strategic objectives are achieved.

2.2. CUSTOMER RELATIONSHIP

2.2.1. Maintain customer relationships (DC Sense)	• Understand the needs of existing customers and establish relationships with potential new customers.
2.2.2. Sign up contracts (DC Transform)	• Capture customer requirement, and agree on service level targets.
2.2.3. Customer satisfaction survey	• Plan, carry out and evaluate customer satisfaction surveys.
2.2.4. Analyse customers' surveys and develop action plans. (DC Sense, Seize)	• Analyse customer surveys. This process provides insights into breached service agreements.

2.3. FINANCIAL PLANNING

2.3.1. Develop IT Budget	• Develop IT budget for the next financial period. Allocate the required financial resources for optimum benefits.
2.3.2. Financial analysis and reporting	• Analyse direct and indirect costs of product development and providing services. This process provides insights into the underlying costs, and profitability of the business.
2.3.3. Invoicing	• Establish an accurate and clear system of invoicing customers for products and services.

2.4. HUMAN RESOURCE PLANNING	
2.4.1. Identify needed skills and competencies (DC Sense)	• Identify and evaluate the needed skills for the organisation to achieve its goals.
2.4.2. Recruit qualified staff (DC Seize, Transform)	• Define criteria to evaluate and recruit qualified staff. • Recruit qualified staff.
2.4.3. Evaluate and develop staff skills	• Evaluate staff performance and provide feedback. • Define opportunities for staff development.
2.4.4. Plan training needs (DC Seize)	• Analyse the organization's business objectives and process improvement plan to identify training needs. • Develop a plan to deliver training. Select people to be trained. Determine cost and resources.
2.4.5. Deliver and maintain staff training (DC Transform)	• Outsource or deliver training internally; document staff training; assess training effectiveness.

2.5. KNOWLEDGE MANAGEMENT	
2.5.1 Develop KM infrastructure	• Establish knowledge management infrastructure to facilitate the exchange and use of knowledge.
2.5.2 Establish a network of experts (DC Sense)	• Establish a network of internal and external knowledge experts to provide staff on-the-job training and contribute to the development of the knowledge database.
2.5.3 Capture and Disseminate knowledge (DC Seize, transform)	• Record knowledge items in accordance with an existing classification system. • Develop meaningful data analysis reports • Provide staff access to the system.
2.5.4 Maintain knowledge system (DC Sense, Seize)	• Evaluate and update knowledge items.

[Next Technical Solution Development Capability]

3 TECHNICAL SOLUTION DEVELOPMENT (TSD)

The capability covers the process areas that apply to the lifecycle of the development of a software product or service. The literature review highlighted that this is a highly mature area across all standards.

3.1 SOFTWARE REQUIREMENTS

3.1.1 Obtain software Scope	• This is an input from Scope Management (project charter), which identifies the high-level software objectives as approved by the customer.
3.1.2 Elicit requirements	• Identify fact-finding sources and techniques to elicit requirements. • Develop requirement definition document that aims at the business audience, customers and users.
3.1.3 analyse requirements	• Analyse the needs of relevant stakeholders. • Identify detailed functional and non-functional requirements, design constraint and process constraint. • Develop requirement specification document, develop requirements in technical terms necessary for architectural design.
3.1.4 Trace requirements	• Requirements must be compared for traceability and consistency with users' requirements (traceable and consistent). • A requirement must be traceable to some source such as to an authoritative source, whether a person or document. Each requirement should have a unique identifier allowing the software design, code, and test procedures to be precisely traced back to the requirement.
3.1.5 Validate and prioritise requirements	• Frequently review requirements documents with stakeholders. This validation ensures that requirements accurately reflect users' needs (correct, unambiguous and relevant). • Stakeholders must assign a priority to each requirement (e.g. mandatory, critical, desirable or optional).
3.1.6 develop requirements baseline	• Users agree that requirements are correct and complete. The result is a requirements baseline approved and accepted by the customer. This baseline will be an input to the requirement management process (complete and correct).
3.1.7 Develop test cases	• Develop test cases to verify compliance with requirements (testable).

3.3 SOFTWARE DESIGN

3.2.1 Develop software architecture	• Define the system's architecture (hardware, software and manual operations) • Allocate requirements: every requirement must be allocated to at least one element of the technical architecture. • Define internal and external interfaces: requirements must clearly capture all the interactions with external systems and the external environment so that boundaries are clear. • Produce the architectural design document.
3.2.2 Verify design	• Ensure backward consistency with the requirements documents.
3.2.3 Update test cases	• Update test cases to verify compliance with the architectural design.

3.4 SOFTWARE CONSTRUCTION

3.3.1 Develop coding standards	• Standards for naming conventions
3.3.2 Obtain complete test cases	• A set of tests, test cases (inputs, outputs and test criteria)
3.3.3 Code software product	• Produce the software units defined by the design.
3.3.4 Verify individual software units	• Verify the individual software units against their developed test cases.
3.3.5 Verify the integrated units	• Combine the software units producing the integrated software product and verify the integrated product against their test cases.

3.5 SOFTWARE TESTING

3.4.1 Apply system performance testing	• Compare the integrated system with the non-functional requirements. This may include security, reliability, accuracy, speed and usability.
3.4.2 Validate the system	• Run the system in the actual environment or use a simulated test environment and provide users access to assess the system's functionality.
3.4.3 Perform regression testing	• Apply regression testing on modified requirements, design and code.

[Next Project Management Capability]

4 PROJECT MANAGEMENT

This capability adopts the practices needed to plan, monitor & control and close the project by mobilising the necessary human and non-human resources and acquiring additional expertise if needed, to deliver the product on time within the agreed upon budget, scope and quality.

4.1 PROJECT INITIATION

4.1.1. Perform feasibility analysis (DC Sense, Seize)	• Define project goals • assess operational, schedule, technical and financial feasibility • develop business case
4.1.2. Initiate scope	• develop high level scope
4.1.3. Initiate schedule	• develop initial schedule
4.1.4. Initiate budget	• develop cost estimates
4.1.5. Initiate communication management	• develop key stakeholders' register • develop stakeholder management strategy • identify information needed for project stakeholders including sponsors and team members
4.1.6. Initiate team	• select project manager • identify required skills (project level) • assemble project team
4.1.7. Initiate risk analysis (DC Sense)	• carry out initial risk assessment • Identify risks
4.1.8. Initiate supplier management (DC Sense, Seize, Transform)	• Identify potential aspects of product that can be acquired by third party supplier • identify potential suppliers • obtain proposals • Negotiate contract terms • sign supplier contract

4.2 PROJECT PLANNING

4.2.1. Plan scope	• develop scope document. • create detailed work break down structure.
4.2.2. Plan schedule	• assign resources. • create schedule. • identify major milestones. • develop schedule baseline. • update project documentation.

4.2.3. Plan cost	• estimate cost • determine budget • develop cost baseline
4.2.4. Plan quality	• identify quality metrics identify checklists • plan quality inspections • develop Quality Management Plan • Update project documentation
4.2.5. Plan communication	• develop Project Communication Plan • plan team access to information • plan project reporting to stakeholders • plan team progress review meetings • plan status reporting (product, risks, quality) • plan supplier status reporting • update project documentation
4.2.6. Plan human resources	• plan team recruitment • assign roles and responsibilities • identify procedures to manage HR conflicts • plan team development • plan user training
4.2.7. 4.2.8 Plan risks (DC seize)	• develop Risk management plan • develop risk avoidance actions • develop contingency actions • update risk register
4.2.8. Plan supplier management (DC Seize)	• develop procurement management plan • develop statement of work • plan supplier review meetings

4.3 PROJECT EXECUTION (DC Transform)

4.3.1. Execute scope	• execute scope document • approve requirements baseline • update scope document
4.3.2. Execute schedule	• update schedule tracking sheets
4.3.3. Execute cost	• update cost tracking sheets
4.3.4. Execute quality management plan (DC Sense, Seize, Transform)	• execute quality management plan • conduct quality reviews • analyse metrics and develop action plans
4.3.5. Perform communication management (DC Sense, Seize, Transform)	• implement team access rights to needed information • conduct team progress review meetings • collect project status reports (product performance, risks and quality) • collect supplier status reports • update project documentation

4.3.6. Execute human resource management	• acquire team (project level) • resolve HR conflicts • conduct team training • conduct user training
4.3.7. Execute risk management plan (DC Sense, Seize, Transform)	• execute risk management plan • conduct risk review meetings • evaluate and quantify risks
4.3.8. Perform supplier management (DC Sense, Seize, Transform)	• conduct supplier review meetings • collect supplier deliverables

4.4 PROJECT MONITORING & CONTROLLING (DC Sense, Seize, Transform)

4.4.1. Integrate change control	• control project changes including requirements • conduct change review meetings • assess change requests • approve/ reject project changes • recommend corrective actions
4.4.2. Monitor and control scope	• verify scope document
4.4.3. Monitor and control schedule and cost	• model and analyse measurements • monitor deviations from baselines. • assess impact of deviations on project • recommend and implement corrective actions
4.4.4. Monitor and control cost	• model? and analyse measurements • monitor deviations from cost baseline • assess impact of deviations on project • recommend and implement corrective actions
4.4.5. Monitor and control quality	• analyse compliance and non-compliance reports • assess impact of non-compliance on project • recommend and implement corrective actions
4.4.6. Monitor and control communication	• Produce and distribute project reports to key stakeholders • review communication plans • monitor update of project documentation • recommend and implement corrective actions
4.4.7. Monitor and control human resources	• analyse team performance reports • assess impact of poor performance on project • recommend and implement corrective actions
4.4.8. Monitor and control risks	• analyse risk reports • assess impact of risks on project • recommend and implement corrective actions
4.4.9. Monitor and control supplier	• audit supplier deliverables • accept/reject supplier deliverables.

4.5 PROJECT CLOSING

4.5.1. Close supplier contract	• Supplier agreements have been verified and completed
4.5.2. Update project documentation (DC Sense, Seize)	• Project documentation completed and updated: ○ Identify lessons learned ○ Update organisational assets

5 SUPPORT CAPABILITY

This study views the maintenance of the software as a separate capability that performs all the activities required to support the product during post-delivery operation. Latest research studies stressed the need for software organisations to treat the support activities after the product being released as a service providing value to their customers.

5.1. SERVICE LEVEL MANAGEMENT

5.1. Identification of Service Requirements	• Capture desired outcomes (requirements from the customer viewpoint) for new services or major service modifications
5.2. Agreements sign-off	• Identify service level targets for customers and sign agreements.
5.3. Service level monitoring and reporting	• Monitor achieved service levels and compare them with agreed service level targets. This information is the basis for measures to improve service quality.

5.2. CHANGE MANAGEMENT

5.2.1. Review of request for change	• A Change Proposal describes a proposed major change. The purpose of Change Proposals is to communicate a proposed major change and assess its risk, impact and feasibility.
5.2.2. Assessment of change request	• Assessment of change proposals typically submitted for significant changes.
5.2.3. Change planning	• Authorize detailed software development project planning, and assess resulting Project Plan prior to authorizing implementation of change.
5.2.4. Change deployment authorisation	• Assess if all required change components have been implemented and properly tested, and authorise change deployment phase.
5.2.5. Post Implementation Review and Change Closure (DC Sense, Seize)	Assess course of change implementation and achieved results and make sure that mistakes are analysed and lessons learned before closure

5.3. Release MANAGEMENT

5.3.1 Release planning	• Define scope and content of Releases. • Develop a schedule for building the infrastructure, deploying and testing the Release.
5.3.2 Establish release infrastructure	• Develop infrastructure on which software will run. Release infrastructure covers hardware, storage, network connections, bandwidth, software licenses, user profiles and access permissions.
5.3.3 Release documentation	• Document released components • Record dependencies among the components
5.3.4 Deploy release	• Deploy release as per developed documentation • Improve documentation if needed.

5.4. CONFIGURATION MANAGEMENT (DC Enabler)

5.4.1 Identify configuration items	• Identify configuration items and baselines • Define who is authorized to make certain changes to the CMS
5.4.2 Establish configuration management system	• Set up change control board whose primary function is to approve or reject all change requests sent against any baseline
5.4.3 Control configuration items	• Any modifications to configuration items approved and adequately recorded in configuration management database.
5.4.3 Perform Configuration Audits	• To perform regular checks, ensuring that the information contained in the CMS conforms to what is actually installed in the live production environment.

5.5. PROBLEM MANAGEMENT

5.5.1 Proactive problem identification	• This process has been added to ensure proactive Problem Management such as improvement of unreported bugs and security patches.
5.5.2 Problem categorization and prioritization	• Allow prioritisation of problems
5.5.3 Problem diagnosis and resolution	• Identify underlying root cause of a problem and initiate the most appropriate and economical problem solution
5.5.4 Problem closure and evaluation (Seize)	• Ensure that, after a successful problem solution, problem is documented and that related knowledge management records are updated.
5.5.5 Major problem review (DC Sense)	• Review history of major problems in order to prevent recurrence and learn lessons for the future
5.5.6 Problem Management Reporting	• Ensure that customers, related parties and service level management are informed of resolved as well as outstanding problems and their processing

<< End of Information Sheet>>

11.6 Second Round Feedback Sheet

Software Process Capability Model for Small Software Enterprises

Objectives of Round 2

The main objective of round 2 is to allow you and other panellists to reassess your initial judgments about your rating of the processes in the previous round.

In this document I have provided information/ summaries of panellists' responses on the different sections of the questionnaire completed in round 1 (in slightly edited form). A column chart is provided for each capability and process that shows the total number of votes on the 5-point Likert scale (0-4).

Changes in the Model

After thorough consideration of panel members' comments and Likert scale voting, the model has been entirely reassessed and has undergone moderate changes. Some processes have been combined with other processes, moved to different capabilities or removed based on panellist recommendations. The project management capability has been changed and new deliverable oriented processes were considered.

A recommendation has been advanced by a number of panel members who suggested classifying the practices into Basic or Advanced levels. In this round you are asked to classify the processes into basic or advanced practice.

Moreover, the knowledge management process has been removed from the model. The quantitative analysis of responses showed obvious lack of agreement among panellists on the feasibility of knowledge management for small software enterprises (SSEs).

In this document I have provided comprehensive feedback of panellist comments on the five capabilities and provided my own comments to clarify important unclear ideas underpinning the research and the proposed model.

Important Objectives to Keep in Mind!

I ask you to attempt the survey with reference to this feedback sheet. When completing the survey, please keep in mind the following important objectives:

- We are attempting to provide small software enterprises (SSEs) with a framework of recommended processes and sub-processes.
- The model provides a common structure and defines the processes needed by SSEs involved in the development and maintenance of software.
- The set of processes and activities can be adapted according to the software project or organisation needs.
- Organisations are not obliged to implement the entire model. They can choose focus areas and plan to implement parts of the model gradually.

Part I - Demographic Information

Following is the demographic information about participants (Questions 1-8):

		# of Participants
2. GEOGRAPHIC LOCATION	North America	3
	South Asia	2
	Australia and New Zealand	4
	Total	***9***
3. NUMBER OF EMPLOYEES	1-10	1
	10-30	1
	30-50	1
	>50	3
	Unidentified	3
	Total	***9***
4. YEARS OF EXPERIENCE	5-10 years	3
	10-25 years	4
	25-50 years	2
	Total	***9***
5. ROLES ASSUMED	Software Process Improvement	4
	Software Quality Audits	4
	Quality Assurance	5
	Project Management	4
	System Analyst	2
	Solution Architect	
	Software Development	4
	Software Testing	2
	Release and Deployment	2
	Configuration Management	1
	Software Maintenance	1
	Others	4
6. CERTIFIED SPI CONSULTANTS	Holders of SPI Certifications	5
7. QUALIFICATION	Undergraduate	5
	Post Graduate	4
	Total	***9***
8. MONETARY VALUE OF PROJECTS	Less than $50,000	
	$50,000 – $100,000	1
	$100,000-$500,000	4
	$500,000-$1000,000	3
	more than $1000,000	1
	Total	9
9. EXPERIENCE IN SSEs	Yes	***5***

Panellists' General Comments on the Model

- In general, as a practitioner, I feel the methodology is too robust and structured to be useful or practical for a small software enterprise. I think there are a lot of relevant processes and sub-processes, and must-haves but generally they need to be scaled back.
- I think my general comment is the methodology is far too structured for a SSE to be able to practically adopt. Why not develop a methodology that shows all the must-haves, and differentiate those from the processes and sub-processes that are optional (resource and time permitting). I would also simplify the sub-processes, consolidate or combine them.

Important points raised by panellists on the proposed model:

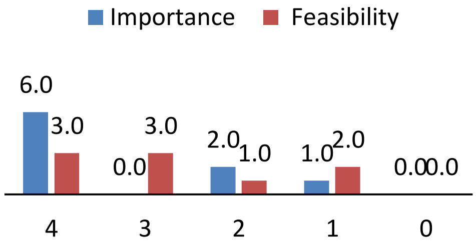
PANELLIST COMMENTS	RESEARCHER COMMENTS
✦ For a holistic approach all capabilities are important. However, depending on the maturity of the organisation with respect to process, people, project management and solution development practices, the organisation has to calibrate its focus based on needs/gaps. ✦ I think all of the capabilities are important. However, the degree to which each of these is executed should be adapted to meet the goals of the company and the specific software. I think in an ideal world, we like to think all processes are feasible for a SSE but in reality they usually operate with limited resources. ✦ You may need to state in your model <u>basic</u> practices vs. <u>advanced</u> practices. Small ventures are usually in survival mode therefore, they will not consider the capabilities 1, 2 and 5 as essential.	It has not been intended that the model should provide an exhaustive set of practices that must be implemented entirely to achieve the desired results. Organisations can select focus areas to start with and then gradually implement more practices. Applying the classifications of basic and advanced at the end of round 2 will improve the model in this respect.
✦ In my view, project management and technical solutions (except for design) would be assigned the highest priority, then change management. Design is usually an advanced practice. All of the other practices are useful, just not easy to use without basic practices in place first . ✦ I see 1.Process Definition to be the lowest priority for SSEs. I don't necessarily agree but their bread and	It is important to provide a common definition for basic and advance practices. The application of basic processes is fundamental practices that lay down the foundation for the advanced processes. SSEs that institute processes at a basic level are able to deliver more successful projects by meeting customer expectations of time, budget and quality. Advanced practices can be implemented once the organisation has achieved confidence with the basic processes.

butter is to develop and sell software. All the nice-to-haves processes will fall by the way-side.	Organisations that implemented advanced level processes have standardised, documented, measured and continuously improved processes; they are very likely to achieve better organisational performance and are able to respond fast and effectively to customer, market and technology changes.
✦ Some panellists commented that the technical solution capability feels like a waterfall model, whereas software organisations are likely to use Agile/Scrum.	The proposed model can be implemented with any software development lifecycle model of choice including Agile/Scrum.
✦ The information sheet mentioned "the model will help SSEs meet the requirements of multiple standards." The practices help a lot, but do not fully implement any one of the other standards listed (ISO/IEC 15504 and CMMI). ✦ The motivation that people will have to use your model won't be to "meet international standards". It will be to "fix my deadline or budget problem".	The main outcome of this research study is a process reference model that will guide small software organisations in their process improvement initiatives. The model should: **2.** provide the needed processes to help organisations meet the perceived project outcomes **3.** build upon multiple software engineering standards **4.** cover different aspects of software organisation without the need to implement multiple SPI standards **5.** be informed by a dynamic capability approach **6.** be feasible for application in small software organisations.
✦ The Dynamic Capability approach adds an unneeded level of complexity; it adds another level of academia. ✦ The need for DCs depends on the market dynamism; in highly competitive markets the DCs approach will help. However, there is a need to assess benefits against cost . ✦ The DC approach adds complexity to the organisation. However, it is important if they want to expand their business and manage changes .	Dynamic Capabilities can be criticised as being very academic; rather, they consist of many well-known processes such as alliance, knowledge acquisition and strategic decision making. In moderately dynamic markets organisations require efficient and robust processes. In contrast, in high-dynamic markets DCs rely extensively on new knowledge created for specific situations. Some firms will be stronger than others in performing some or all of these tasks DC processes should be relatively simple to allow for emergent adaptation, but should not be completely unstructured.

Part II – Five Capabilities/Process Groups

1. PROCESS DEFINITION AND IMPROVEMENT

6 out of 9 panellists gave importance a ranking of (4). 6 members gave feasibility a rank of (4) or (3). Two members regarded it as not very feasible (1).

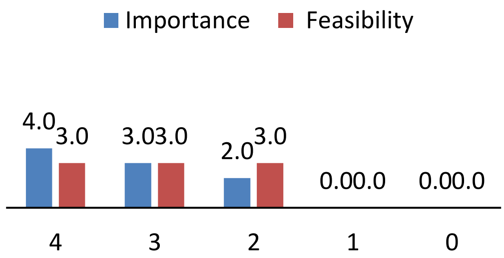


PANELLISTS COMMENTS

- It is critical for SSEs to describe and improve processes if they are to expand and grow without losing control of quality and reputation.
- It is an extremely important capability, but normally not handled well by organisations.
- it can be a challenge for SSEs to dedicate the time and resources to this capability; however, it is feasible for them to do so.
- Processes for this capability must be simple, lean and compatible with agile development methodology, which is very common methodology used by a small or any organisation.

2. ORGANISATIONAL

All responses to Importance and Feasibility were 2 or above. A minimum of three members considered it highly important and highly feasible.

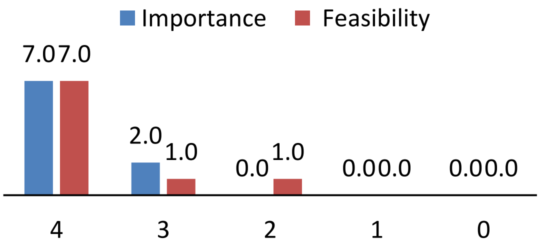


PANELLISTS COMMENTS

- Requires leadership and is less likely to get done well.

3. TECHNICAL SOLUTION DEVELOPMENT

The majority of panellists considered this capability as very highly important and very highly feasible.



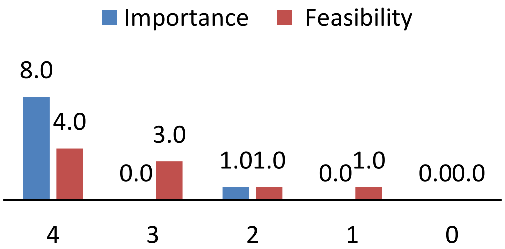
PANELLISTS COMMENTS

- This is a capability that has direct impact on clients and is often demanded by them.

4. PROJECT MANAGEMENT

The majority of panellists rated this capability

as very highly important. There is lack of agreement among you on its feasibility for SSEs.



PANELLISTS COMMENTS

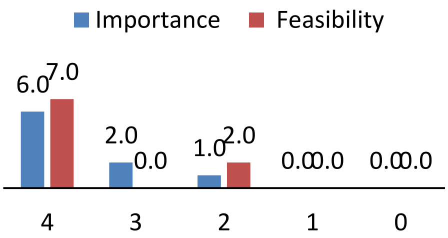
- Probably the most essential set of practices, that might not be seen as essential by a small business.

5. SUPPORT

The majority of panellists rated this capability as very highly important and very highly feasible.

PANELLIST COMMENTS

- This is a capability that has direct impact on clients and is often demanded by them (same comment as above).



DYNAMIC CAPABILITY APPROACH

POSITIVE COMMENTS

- It depends on the market dynamism; in highly competitive markets the DCs approach will help. However, there is a need to assess this against cost.
- I think the approach is relevant. However, the extent to which it is adopted should be practically applied or adapted to meet the software development and the organisation's goals .
- The Dynamic Capabilities (DCs) approach provides the capability to improve and sustain businesses and must be encouraged.
- DCs are needed in software organisations to ensure they understand current gaps and needs of their clients as well as maintaining a proactive stance on emerging needs. This will help them focus their goals and ensure value is rendered where required.
- DCs are needed by SSEs as it will enable them to respond fast to rapid changes in technology, business opportunities and customer needs as they happen.
- I think Dynamic Capabilities are just as important for SSE as they are for larger organizations, and critical in today's highly volatile world. The processes and the roles involved in supporting the DC need not and will not be as complex but the SWOT analysis must be adopted for the SSE to understand and respond to current strengths, weaknesses, opportunities and threats.

NEGATIVE COMMENTS

- Dynamic Capability adds an unneeded level of complexity; it adds another level of academia.
- It could add unnecessary levels of complexity if it is not used or applied to meet a specific objective that can be measured.

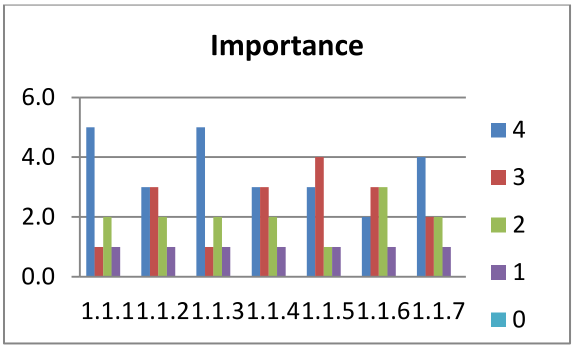
Part II – PD&I Capability

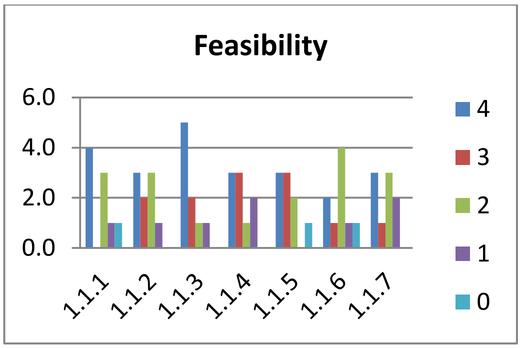
PANELLISTS' GENERAL COMMENTS

- I think this capability is important but is often regarded as an overhead in many small organizations. The key is to identify those processes that are critical in establishing solid practices to sustain the organization over time and that can deliver meaningful insights for continuous improvement.
- A SSE should use minimum standards to develop their critical processes. The challenge for SSEs is 'how deep' and 'how broad' to go with respect to instituting standards and best practices that provide the most value to the organization. Perhaps it would be useful to identify the key processes within the Process Architecture by way of focusing the organizational resources on what is important.
- I would like to emphasize that processes are an integral part of a business solution - the others being people and tools/software. A process change may be triggered by 1) an organizational change (i.e. When functions within an organization are collapsed, removed or added); or 2) the introduction of software that automates previously manual processes. The key is that process changes are not triggered by process improvements alone but also by organizational or software changes.
- I suggest that all the activities under Process Review are DC Sense. All the activities under Process Improvement are DC Transform.

1. Process Definition and Improvement

1.1. PROCESS ESTABLISHMENT (DC Enabler)





PANELLISTS' COMMENTS

1.1.1. Initiate process establishment

- This process establishment is key to ensuring processes are consistently designed, governed and improved over time.
- Process definitions should be 1-2 pages. I would stress that you mean "LITE Process Definition and Improvement".
- Process establishment and improvement is critical for organisational performance. However, SSEs must establish processes that work for them and avoid getting swamped by large cumbersome processes. In my dealings with SSEs the target is always to put "process on one page".

- A small software team doesn't require this initially, compared with the engineering and project management capabilities.

1.1.2. Define a process

- In the words of Deming, "If you can't describe what you are doing as a process, you don't know what you're doing."
- Every process must have consistent attributes, including ownership, identification of objective, measures and target SLAs.
- Should be LITE PRAGMATIC processes, not heavy long processes.
- This is an advanced practice.

1.1.3. Identify work products [combined with 1.1.2]

- Once you have identified the process, it's important to identify the work products to be able to measure outcomes.
- One panel member suggested to rename this process as **' Design Process'**
- This is an advanced practice.

1.1.4. Define process architecture [changed to 1.1.3 'Define interfaces']

- It is important to know where a process feeds into for an overall successful outcome but it's not practically very feasible to include inter-dependencies in processes.
- Process architecture is an advanced practice.

1.1.5. Define control mechanism

- Roles help with PM
- It is feasible and good to have this process.
- One panel member suggested to rename this as 'Define Process Roles & Accountabilities'

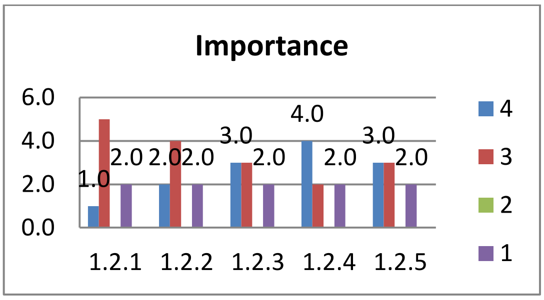
1.1.6. Define process measurements

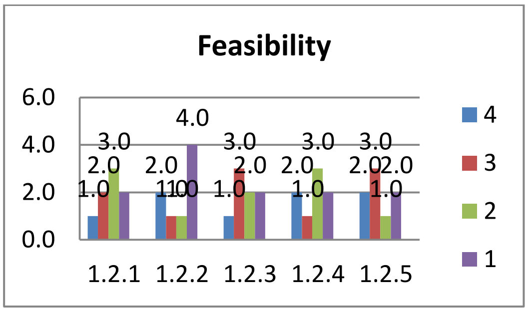
- Very important to demonstrate process relevance for continuous improvement. Unfortunately, sometimes SSEs tend to neglect measurements.
- It's important to have measurements to measure performance. However, it is not feasible to collect measures for all processes and for every project especially when some of the measures are not numeric/tangible.
- Applicable if they stick to 2-3 measures, usually related to PM project monitoring

1.1.7. Document process [removed, will be an outcome of 1.1.2]

- This is an advance practiced and should have lower priority.
- It is very important to retain organizational knowledge and sustain cross-functional learning. However, this is often neglected by SSEs which tend to rely on 'tribal knowledge'
- This is an advanced practice.

1.2. PROCESS REVIEW (DC Enabler)





PANELLISTS' COMMENTS

1.2.1 Plan process review

- It is feasible for SSEs to achieve; however, it will be a challenge to dedicate the time and resources to this process.
- This is an advanced practice and should have lower priority than technical solution and project management.
- It is Important to review and improve but not very feasible due to time and budget constraints.
- In an ideal world, it is an important practice but often not feasible in smaller organizations given the limited resources and competing priorities. For practical reasons, it is something that could be done annually.

1.2.2 Collect and analyse process data (DC Sense)

- This is an advanced practice and should have lower priority than technical solution and project management
- It is Important for measuring process performance but not feasible because it requires a dedicated team to carry this out.
- Not feasible given the limited resources. For practical reasons, it is something that could be done annually.

1.2.3 Review process performance

- This is an advanced practice and should have lower priority than technical solution and project management.
- It is important to identify improvements/opportunities; only feasible if you have the data to conduct the review.
- Not feasible given the limited resources. For practical reasons, it is something that could be done annually

1.2.4 Document review results

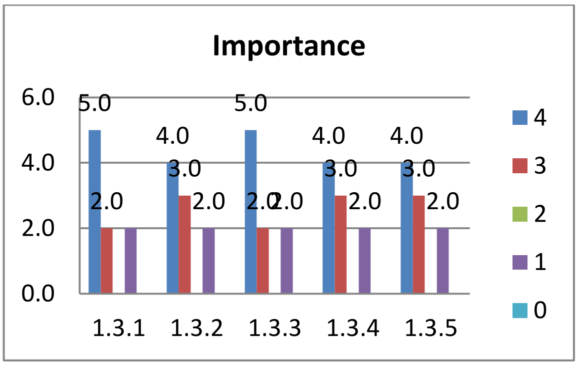
- This is an advanced practice and should have lower priority than technical solution and project management.
- Documentation captures knowledge, which can be a valuable reference for future projects; however, it is not always feasible because it takes time.

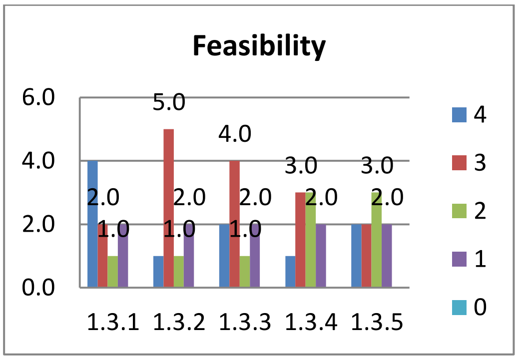
- Not feasible given the limited resources. For practical reasons, it is something that could be done annually.

1.2.5 Report review results

- This is an advanced practice and should have lower priority.
- It's important and feasible because once you have the results, it doesn't take much time to communicate them.
- In an SSE, reporting the results is not very important due to the close communications between teams.
- Not feasible given the limited resources. For practical reasons, it is something that could be done annually.

1.3 PROCESS IMPROVEMENT





PANELLISTS' COMMENTS

1.3.1 Identify and prioritise process improvements (DC Sense)

- It is an Important and feasible process if the organisation knows how to drive it.
- This is an advanced practice and should have lower priority than technical solution and project management.
- Not feasible in small organisations given the limited resources and competing priorities. For practical reasons, it is something that could be done annually.

1.3.2 Plan improvements (DC Seize)

- It can be a challenge for SSEs to dedicate the time and resources to this; however, it is feasible to do so.
- This is an advanced practice and should have lower priority than technical solution and project management.
- Not feasible in small organisations given the limited resources and competing priorities. For practical reasons, it is something that could be done annually.

1.3.3 Implement process action plans (DC Transform)

- This is an advanced practice and should have lower priority than technical solution and project management

- It is important to pilot first to allow the teams to test the process, rather than going for a big-bang approach.

1.3.4 Monitor and evaluate improvements

- This is an advanced practice and should have lower priority than technical solution and project management.
- It is a fairly feasible process but requires defined measures.

1.3.5 Update organisation process asset library.

- This is an advanced practice and should have lower priority than technical solution and project management.
- Important to have the library updated so the latest processes are followed but in practice these repositories do not get updated frequently due to lack of resources and people dedicated for the task.

Part IV ORG Capability

PANELLISTS' GENERAL COMMENTS

- The organizational capability is fundamental to any software company that seeks to run a successful business. The degree and complexity to which it is implemented may vary but the key is that it gets implemented in a way that addresses the basic needs of the organization.
- I have worked in an SSE where rudimentary processes were implemented successfully and matured over time through continuous improvement and through the insight and lessons learned.
- 2.3 Financial planning is a DC Enabler: the better we understand our cost allocations and services, the more insight we have into potential savings / opportunities to improve services to make them more cost effective.
- 2.5.1 Develop KM Infrastructure is a DC Enabler. The exchange and use of knowledge empowers the organization to share, disseminate knowledge and make informed decisions.

Researcher motivation for including Organisational capability

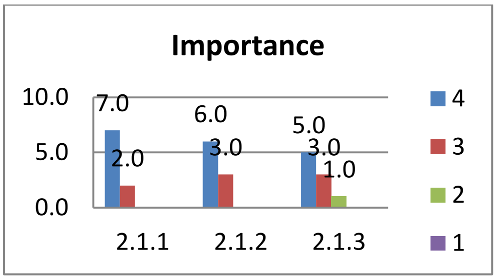
The review of small organisations and small software organisations literature revealed that SSEs are focused on operational effectiveness rather than strategic management. The lack of a strategic planning process has been linked to the poor performance of SSEs. The size of small enterprises allows them to be more adaptable and responsive to changing market conditions and exploiting new opportunities than large organisations.

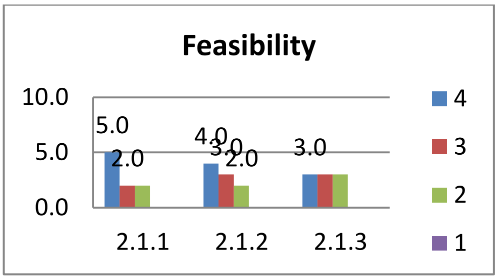
A well accepted study by Lussier (1995) introduced a set of key factors or practices empirically tested for their effectiveness in predicting business success and failure. The findings of the study suggested education and experience of both senior management and staff, the existence of detailed plans, and the use of professional advice, contribute significantly to an organisation's success. Furthermore, the study showed that the owners of failed businesses were not selective in recruiting, selecting and retaining talented employees.

This all suggested the need for the adoption of organisational capability in the model.

2. Organisational Capability

2.1 BUSINES ALIGNMENT





PANELLISTS'GENERAL COMMENTS

- Requires leadership and is less likely to get done well.
- It is important for the company or business to establish clear, measurable goals. It does not have to be complex; therefore it is feasible and necessary for accomplishing good software development.

2.1.1 Identify vision and goals and develop business objectives (DC Sense)

[Changed to 'Develop strategic business objectives']

- Important to understand client needs (current and future) to provide value added services.
- The deliverables of this process are: 1) Identification and prioritization of strategic goals; 2) IT goals to support business objectives/ [strategic goals].
- No business large or small should operate without objectives and strategies.
- Important and feasible because the business depends on it.
- Difficult to do well. Needs leadership audience.

2.1.2 Identify business outcomes to achieve strategic objectives (DC Seize)

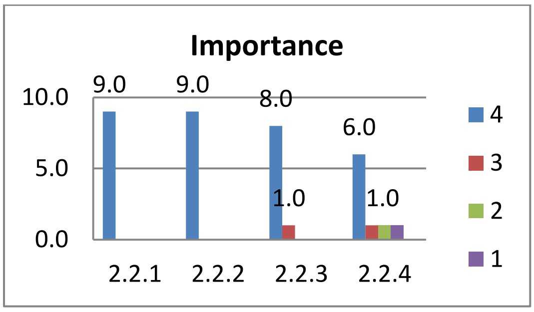
- The deliverables of this process are: 1) identification of programs/projects; 2) prioritization of these; 3) client approval / sponsorship and funding.
- Important and feasible, difficult to do well as it needs leadership audience.

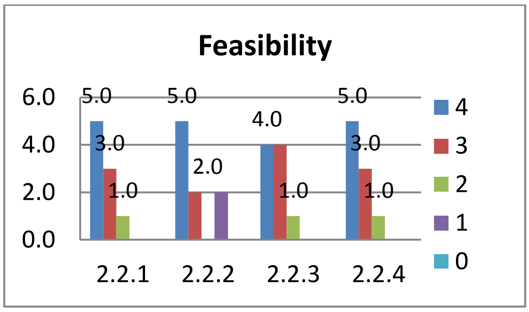
2.1.3 Achieve strategic IT plans (DC Transform)

[Changed to 'Implement strategic plans]

- Important to assess achievement of goals by evaluating realization of anticipated business benefits for each initiative - i.e. plan vs. actual investment vs. business benefit.
- Important and feasible.
- Difficult to do well. Needs leadership audience.

2.2 CUSTOMER RELATIONSHIP





PANELLISTS' COMMENTS

- Requires leadership and less likely to get done well
- Without customers, the company would fail. If the company is not able to execute against these processes, the company would fail.
- Important and feasible through on-going engagement with customers; need to be managed and continuously improved.

2.2.1 Maintain customer relationships (DC Sense)

- Essential to any business, particularly SSEs.
- Important to maintain a client relationship and a proactive posture to be able to anticipate and meet current and future needs. Understanding the client needs and direction is a competitive advantage. This process could be less formal in SSEs - e.g. monthly meetings between the business and IT managers and an annual report to coincide with the annual planning process.
- Very important and feasible

2.2.2 Sign up contracts (DC Seize and Transform)

- Equally important is agreeing on the services and service level targets to manage client expectations and potential resource constraints which many SSEs face.

2.2.3 Customer satisfaction survey (DC Sense)

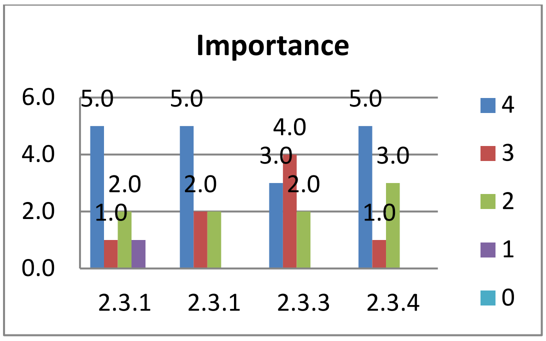
[Changed to 'Carry out customer satisfaction survey]

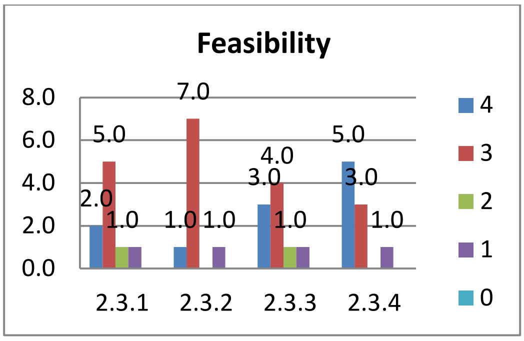
- Surveys are typically employed in larger organizations. However, simplified versions are also a useful and practical tool to elicit client satisfaction or dissatisfaction. This is a process that could be folded into 2.2.1 Maintain Customer Relationships.

2.2.4 Analyse customers' surveys and develop action plans

(DC Seize and Transform)

2.3 FINANCIAL PLANNING





PANELLISTS' COMMENTS

- All these are relevant but they should be simplified to the top critical three items that should be considered in the process.
- Financial planning process is a DC Enabler: the better we understand our cost allocations and services, the more insight we have into potential savings/opportunities to improve services to make them more cost effective.
- Essential to any business to address Financial Planning; the audience is the CFO.

2.3.1 Financial planning

[Removed from the model]

2.3.1 Develop IT budget

- Basic and is required for all organizations but may vary in complexity depending on size of client portfolio, diversity of services and maturity of DC capability.

2.3.1 Financial analysis and reporting

[Changed to 'perform financial analysis and reporting']

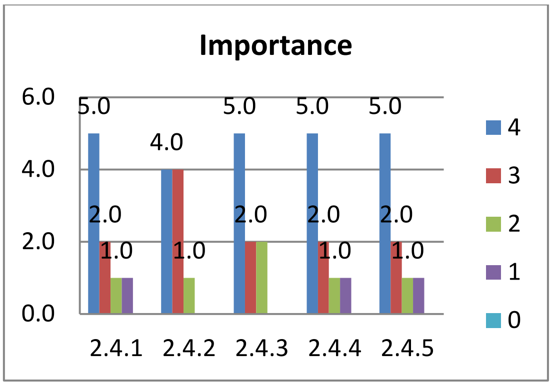
- Large organisation should regularly analyse and report financial situation. SSEs are probably more in touch with their expenditure so this is not as important.

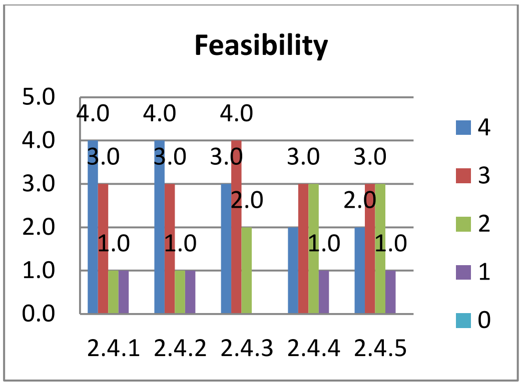
2.3.1 Invoicing

[Changed to 'Establish invoicing system']

- Same as 2.3.1
- A 'must'
- Critical for every organisation large or small!

2.4 HUMAN RESOURCE PLANNING (DC Enabler)





PANELLISTS' COMMENTS

- In an ideal world, all of these components are relevant and important. In real life, the most important thing the company has time and money to do is likely to be identifying the skills required, and recruiting qualified staff to fulfil that skill need (#12).
- Nice to do. Can be delayed, being an advanced practice

2.4.1 Identify needed skills and competencies (DC Sense)

- Talent management and professional development are more formally pursued in big organizations and more informal in smaller SSE. Sometimes, the budget does not allocate for training. My experience in SSE has been that I learned on the job because there was either no budget or no 'convenient time' due to resource constraints.
- Overall, having the right skills and competencies is critical and this applies to all Human Resource Planning.

2.4.2 Recruit qualified staff (DC Seize)

2.4.3 Evaluate and develop staff skills (DC Transform)

- May not be feasible due to time and budget constraints.
- Whilst it is important, SSEs have to balance staffing against training. They should do it as the need arises and they have staff availability.

2.4.4 Plan training needs

- It can be difficult for SSEs to plan, deliver and maintain staff training if they do not have the backup resources to complete the "paying" work whilst staff are away being trained.

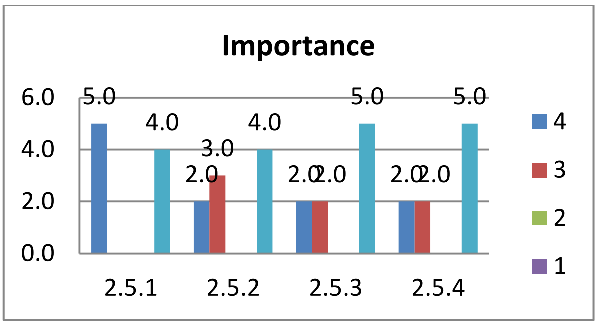
2.4.5 Deliver and maintain staff training

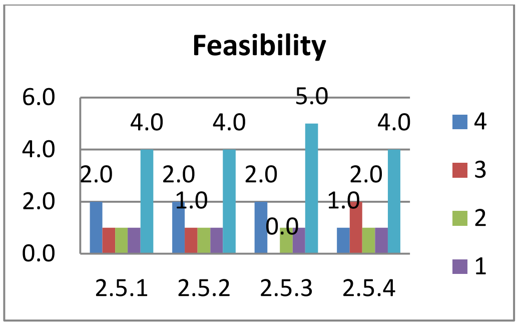
- High staff turnover [very likely in SSEs], tight budget and schedule constraints make it infeasible.

2.5 KNOWLEDGE MANAGEMENT [Removed]

RESEARCHER'S COMMENT

This process has been removed from the model due to lack of consensus among panel members on its feasibility for SSEs. Two panel members recommended the removal of this process.





PANELLISTS' COMMENTS.

- Knowledge management to capture and disseminate organizational knowledge is so important and too often neglected. Dismissed as 'just documentation that someone will eventually get to', it tends to expose an SSE to tribal knowledge and to the risk of lost knowledge when experienced staff leave. In my opinion, knowledge management creates a foundation for instituting consistent practices and fostering learning and growth of the staff.
- Losing corporate knowledge when personnel leave can have a huge impact on a company particularly SSEs.
- Ideally, it is important. In reality, it would likely be the least important of all considerations

2.5.1 Develop KM infrastructure

- Develop KM Infrastructure is a DC Enabler. The exchange and use of knowledge empowers the organization to share, disseminate knowledge and make informed decisions.
- It is rarely implemented in SSEs.

2.5.2 Establish a network of experts (DC Sense)

- Rarely in SSEs. It's usually a staff initiated initiative.
- No real need for a network of experts for SSEs as the smaller working environment is a de facto network of experts.

2.5.3 Capture and disseminate knowledge (DC Seize)

2.5.4 Maintain knowledge system (DC Transform)

- Ensure you include something on removing "old" knowledge that is no longer needed.

- Under Knowledge Management include "Evaluate use of the Knowledge System". I've seen many Knowledge Systems that are not being used by staff.

Part III - Technical Solution Development

Panellists' General Comments:

- We need to think of requirements in levels of granularity. For example, you start with high level requirements, which evolve into use cases, then technical specifications, then design specifications and ultimately test cases. I feel the perspectives and levels of granularity are not adequately addressed in your model.
- I would try to align your process abstract name (e.g. 3.1.3 Analyse Requirements) with your description (e.g. 3.1.3 Develop Technical Specifications) to avoid confusion.
- I would like to add an important practice - 'code reviews'.

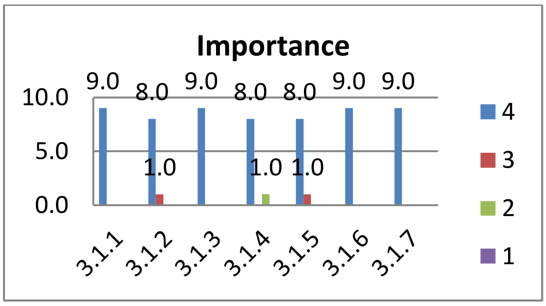
RESEARCHER COMMENTS

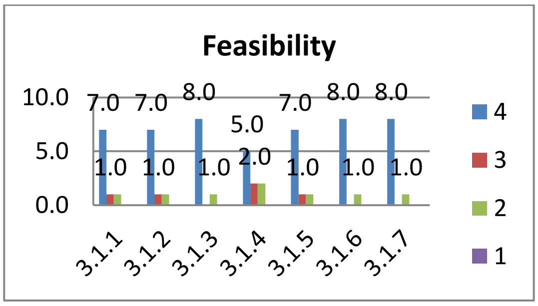
The model already addresses the different levels of requirements through 3.1.1 Obtain software scope which determines the high-level requirements at project initiation, then 3.1.2 Elicit requirements which develops use cases (user perspective of requirements), then 3.1.3 Develop Requirements Specification which aims at developing the detailed requirements. 3.1.2 and 3.1.3 can be handled with light- weight documentation when following agile methodology with emphasis on empowering team collaboration and prototype development for testing the design.

As for code reviews the model already included verification of individual units and integrated units. This practice can use dynamic or static testing including code reviews at individual and team level.

3. Technical Solution Development

3.1 SOFTWARE REQUIREMENTS





3.1.1 Obtain software scope

- A 'must'

3.1.2 Elicit requirements

3.1.3 Analyse requirements [Changed to 'Develop requirements specification']

- Based on description of this process, you may want to rename it to 'Develop technical specifications' as opposed to 'Analyse requirements'.

3.1.4 Trace requirements [Removed and combined with 3.1.5]

- One panel member recommended combining 3.1.4 with 3.1.6.

3.1.5 Validate and prioritise requirements [Changed to 3.1.4]

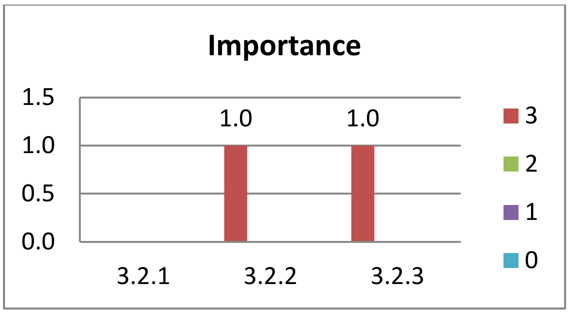
3.1.6 Develop requirements baseline [Changed to 'Establish requirements baseline']

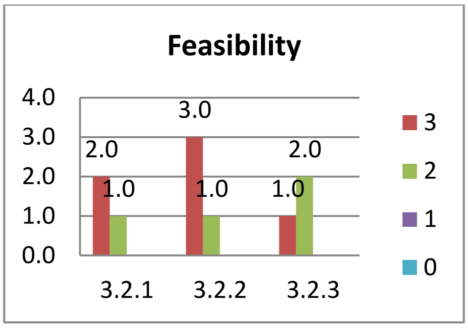
- One panel member suggested combining 3.1.4 and 3.1.6.

3.1.7 Develop Test Cases

- With the number of tests that could be conducted and with the knowledge that each software development initiative is slightly different, it may be useful to have a process called 'Define Test Strategy', which outlines the test environment.

3.2 SOFTWARE DESIGN





3.2.1 Develop software architecture

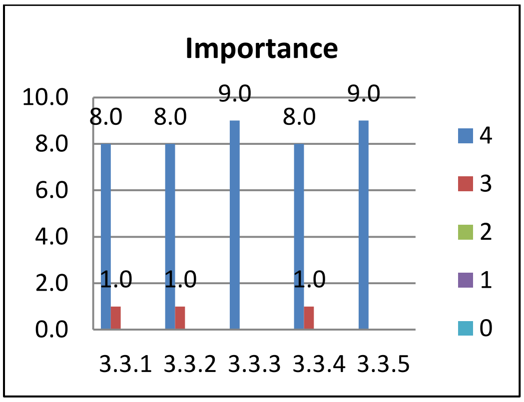
3.2.2 Verify design

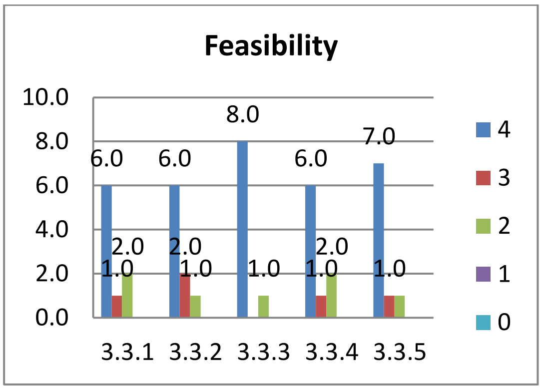
- [Design can be verified using a prototype when following Agile lifecycle model]

3.2.3 Update test cases

- Most times test cases are not updated due to time constraints and an ad-hoc approach is adopted.

3.3 SOFTWARE CONSTRUCTION [changed to 'Software Construction and Testing]





3.3.1 Develop coding standards

- Important for code comprehensibility but it's hard to bring all developers in the team to code in same style and standards as everyone has their own specific way.

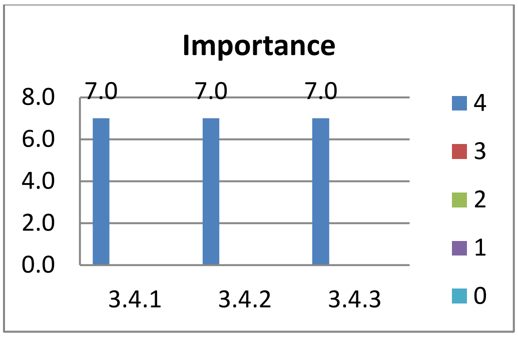
3.3.2 Obtain complete test cases

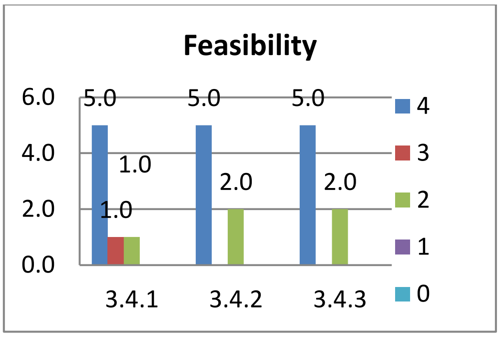
3.3.3 Code software product

3.3.4 Verify individual software units [Unit testing]

3.3.5 Verify the integrated units [Integration testing]

3.4. SOFTWARE TESTING [changed to 'Functional Software Testing']





PANELLISTS' COMMENTS

- In your model there is no mention of data modelling, information architecture data conversion testing. Yet data/information is fundamental to any software development. This should be addressed in the model.
- There is also a volume and load balance test to validate throughout and performance with actual data.
- Some companies also conduct data conversion testing, particularly for software where the data is ported over from legacy systems.

3.4.1 Apply Testing [Changed to 'Apply Performance Testing']

Explanation: Apply non-functional testing in accordance with testing strategy. This may include volume, load balance, performance, data conversion and data model testing.

3.4.2 Validate the System *[user acceptance testing]*

3.4.3 Perform regression testing * [Changed from 3.4.3]

- For small changes, a full suite of regression testing is not always done because it pushes project timelines unless the regression testing is automated.

Part IV PM Capability

Panellists' General Comments:

- The capability is very important for SSEs, the activities are too detailed and should be collapsed. It is important not to lose sight of the essence and the value by adding too many activities.
- All these practices are important but could be collapsed into more practical terms. For example -

  4.1 Project Initiation

  4.1.1 Develop Business Case / Feasibility Study

  4.1.2 Develop Project Proposal (Includes high level project goal, scope, schedule, cost, risk analysis, project resource requirements based on skill set)
- Reasonably good. Critical for SSEs is not to burden them with too much process so whilst in a large organisation, these processes can be time consuming and large in nature, SSEs should look to what I call "Planning on a Page" (or maybe 2).

Recommendation for additional processes

- The project management capability 'Project Execution' activity must include all the preparatory and transitional activities required to cut-over to Production, such as engagement of support team, training, define and sign-off Operating agreement. I disagree with your model where it currently sits in the 'Support Capability'. Support implies production and operations. The project management capability is in essence delivering a solution in production and the 'Project Close' must include the implementation of the solution in that environment as well as the post-implementation review.

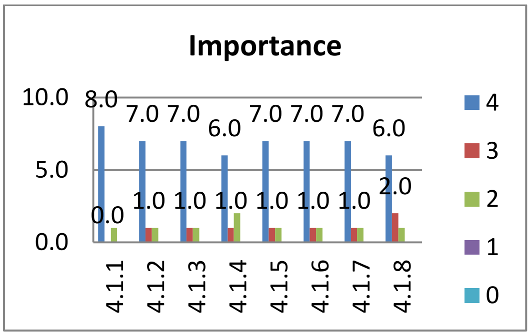
RESEARCHER COMMENTS

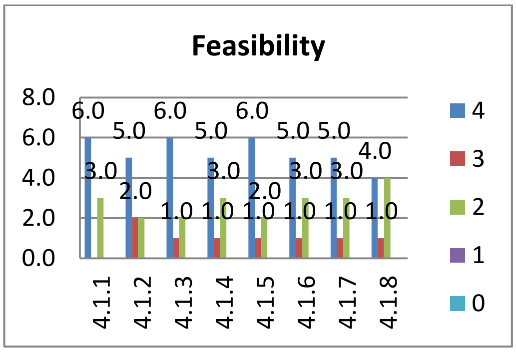
IMPORTANT NOTE:

- Panellists' comments stressed that this capability can be structured using PMBOK process groups initiating, planning, executing, monitoring & controlling and. Therefore, the PM capability has undergone major changes, resulting in a simpler more practical processes aligned with PMBOK processes.

4.PROJECT MANAGEMENT CAPABILITY

4.1 PROJECT INITIATION





4.1.1 Perform feasibility analysis (DC-Sense)

4.1.2 Initiate scope (DC Sense)

4.1.3 Initiate schedule (DC Sense)

4.1.4 Initiate budget (DC Sense)

4.1.5 Initiate communication management (DC Sense)

4.1.6 Initiate team (DC Sense)

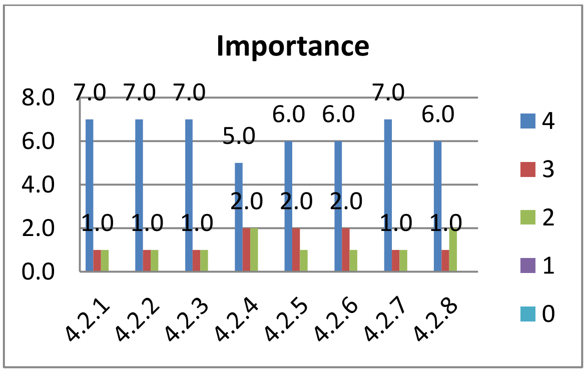
✦ At this stage, a PM and a Project Business Sponsor may be selected but team cannot be assembled until project resource requirements are established (4.2.6).

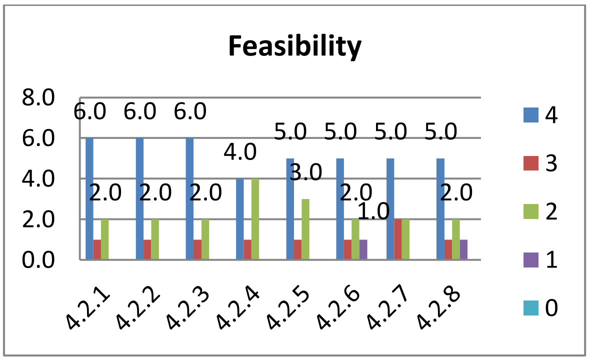
4.1.7 Initiate risk analysis (DC Sense)

4.1.8Initiate supplier management (DC-Sense)

✦ At this stage, this practice should include only the identification of potential suppliers and the request for proposals, not any contractual agreement, as detailed requirements are not clearly defined yet.

4.2 PROJECT PLANNING





4.2.1 Plan scope

✦ Again, all these practices are important and are essentially an elaboration of the project initiation practise. This should be emphasized to maintain the integrity and simplicity of your model.

4.2.2 Plan schedule

4.2.3 Plan cost

4.2.4 Plan quality (DC Seize)

✦ Quality is achieved by validating, prioritizing, tracking requirements and alignment to design. Not sure those SSEs will require any complex quality metrics or checks that would not already be achieved through effective requirements management and testing.

4.2.5 Plan communication (DC Seize)

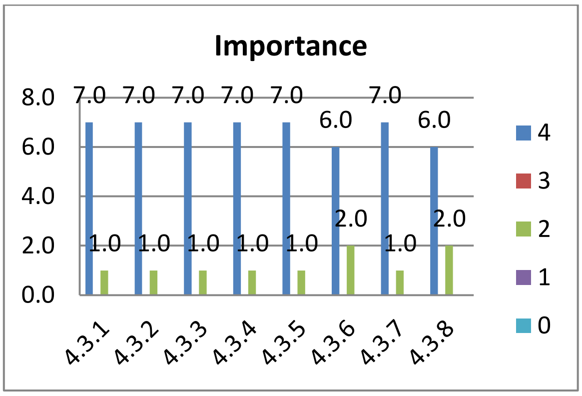
✦ The degree to which each practise is applied will depend on the complexity of the project. However most SSE projects tend to be mid-sized and therefore do not require the full practise to be executed. For example, 4.2.5 can involve simple activities like weekly team meetings and monthly status meetings with the project

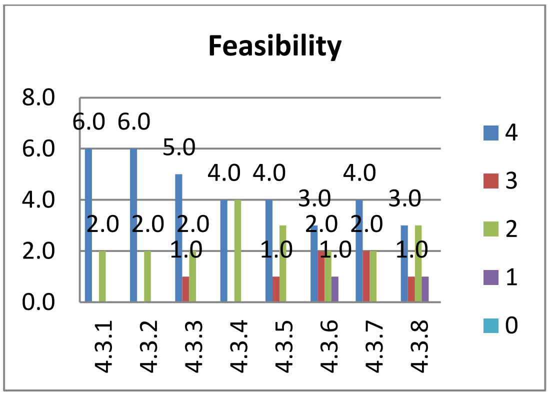
4.2.6 Plan human resources (DC Seize)

4.2.7 Plan risks (DC Seize)

4.2.8 Plan supplier management (DC Seize)

4.3 PROJECT EXECUTION





4.3.1 Execute scope (DC Transform)

✦ A 'must'

4.3.2 Execute schedule (DC Transform)

✦ A 'must'

4.3.3 Execute cost

✦ A 'must'

4.3.4 Execute quality management plan (DC Transform)

- Quality activities may be embedded as part of requirements management and testing on small to mid-size initiatives.

4.3.5 Perform communication management (DC Transform)

- Smaller teams tend to collaborate and communicate more informally and frequently. Formal communication could be simplified through weekly team status and monthly project reporting to the sponsor or steering committee.

4.3.6 Execute human resource management (DC Transform)

4.3.7 Execute risk management plan (DC Transform)

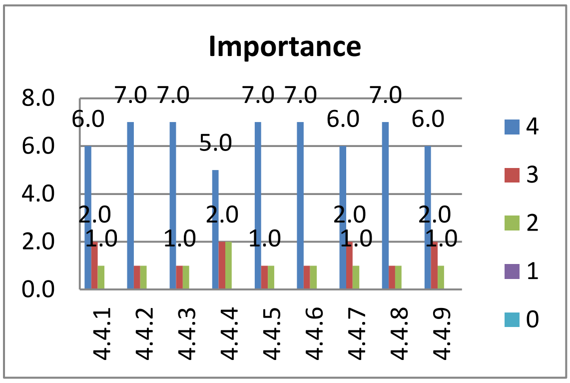
- Degree to which this activity is executed may vary depending on project complexity, scope and cost.

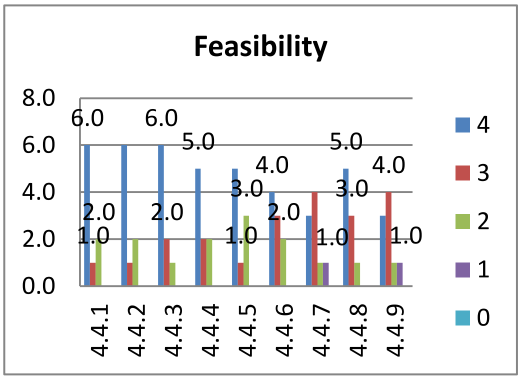
4.3.8 Perform supplier management (DC Transform)

- Degree to which this activity is executed may vary depending on project complexity, scope and cost.

4.4 PROJECT MONITORING & CONTROLLING

- Project monitoring and control is a shadow process that occurs throughout the project's life cycle. As such, the data that feeds into it can be captured in the initiation, execution and close phases. This process can be collapsed into one, simply called 'Project Monitoring and Reporting, focusing on reporting change, scope, schedule, cost, HR, risk and quality.





4.4.1 Integrated Change Control

4.4.2 Monitor and control scope

4.4.3 Monitor and control schedule and cost

4.4.4 Monitor and control cost

4.4.5 Monitor and control quality (DC Transform)

4.4.6 Monitor and control communication (DC Transform)

4.4.7 Monitor and control human resources (DC Transform)

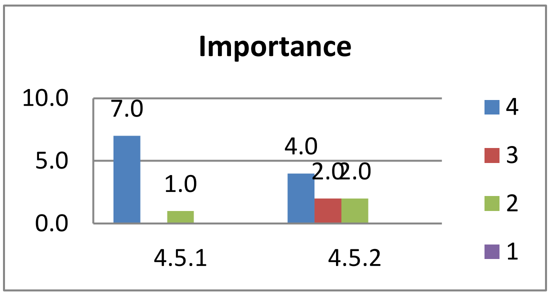
4.4.8 Monitor and control risks (DC Transform)

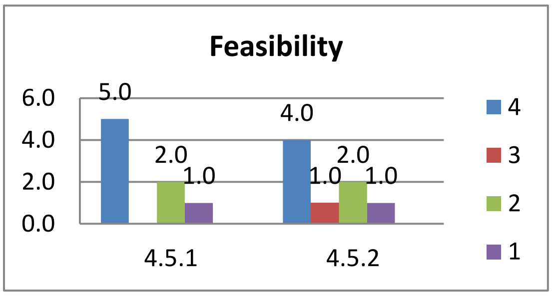
4.4.9 Monitor and control supplier (DC Transform)

4.5 PROJECT CLOSING

- The project management capability 'Project Execution' process must include all the preparatory and transitional activities required to cut-over to *Production*, such as engagement of support team, training, define and sign-off operating agreement. I disagree with your model where it currently sits in the 'Support Capability'.

  Support implies production and operations. The project management capability is in essence delivering a solution in production and the 'Project Closing' must include the implementation of the solution in that environment as well as the post-implementation review.





4.5.1 Close supplier contract

4.5.2 Update project documentation (DC-Sense, Seize, Transform)

Part IV SUPPORT Capability

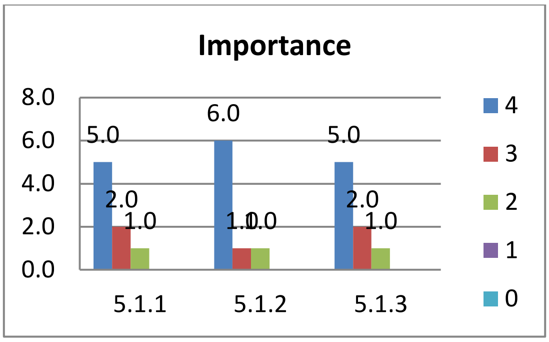
Panellists' General Comments:

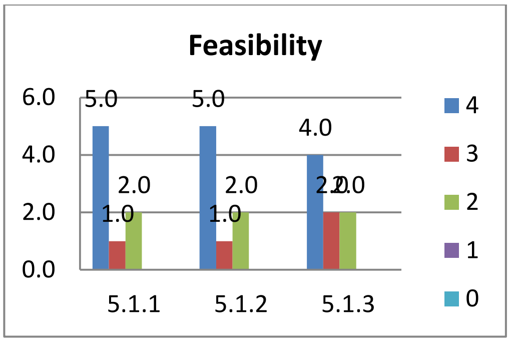
- This capability is important to maintain the ongoing relevance of the software and ensure it continues to meet the business needs. Business ownership for the application and decision/approval rights for major changes should be captured in the configuration DB. At minimum (and for SSEs), the configuration DB must include the application ID, name, business application owner, technical SME, app description, service recovery document, etc. All the above information required to support the application in production should be a deliverable of the project and part of an implementation checklist.

- Important to sustain the products and business and to deliver high quality maintenance.

5 SUPPORT CAPABILITY

5.1 SERVICE LEVEL MANAGEMENT





5.1.1 Identification of service requirements

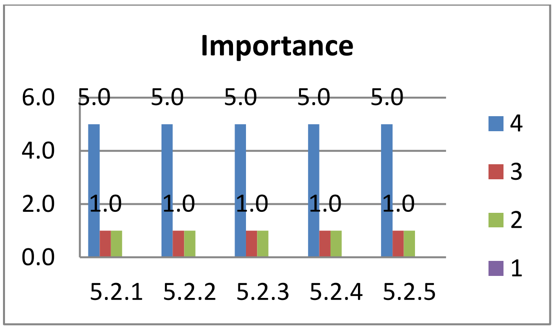
- These are all important but typically not formally conducted in SSEs as the project team members usually become the support team once the software cuts over to production. It is however important to ensure service level targets and client expectations are agreed upon.
- Harder to define

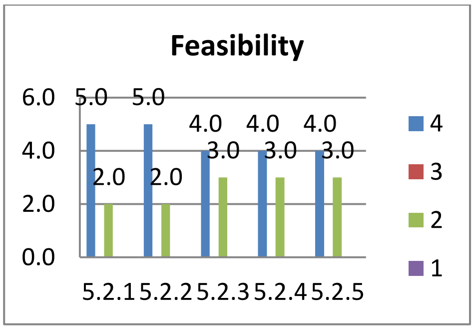
5.1.1 Agreements sign-off

- This is critical for SSEs as many of them will do "extra" services to keep their customers happy and some customers take advantage of that. Agreements can help get the service back to the agreed scope

5.1.1 Service level monitoring and reporting

5.2 CHANGE MANAGEMENT





5.2.1 Review of request for change

5.2.2 Assessment of change request

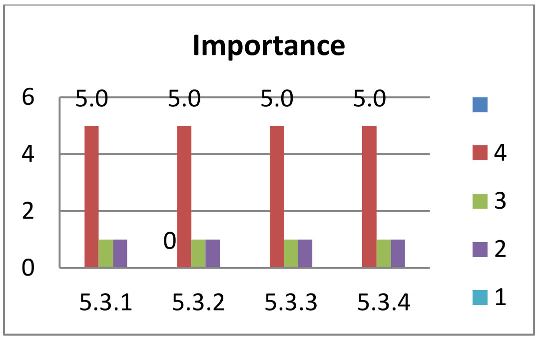
5.2.3 Change evaluation and authorisation

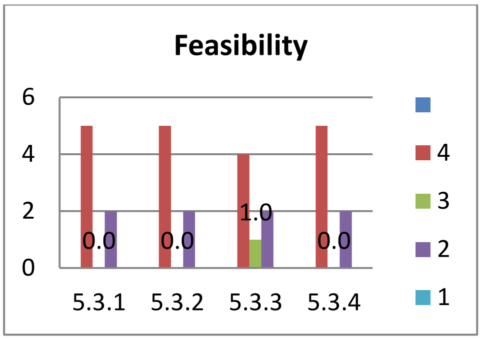
5.2.4 Change deployment authorisation

5.2.5 Post implementation review and change closure (DC Sense, Seize)

5.3 RELEASE MANAGEMENT

[Release management has been moved to PM Closing process]





5.3.1 Release planning

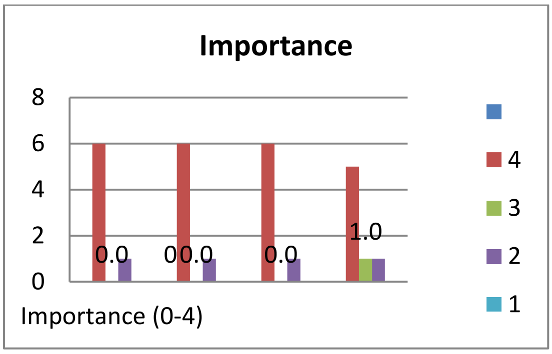
5.3.2 Establishment of release infrastructure

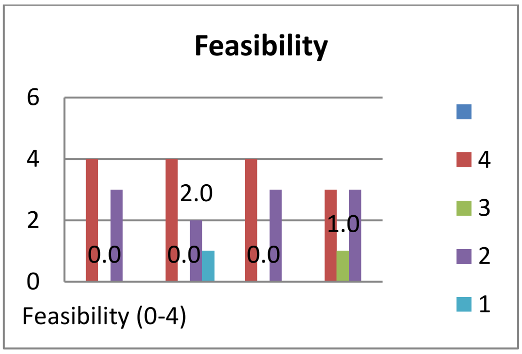
5.3.3 Release documentation

- Oftentimes the documentation quality is not optimal.

5.3.4 Release deployment

5.4 CONFIGURATION MANAGEMENT (DC-Enabler)





PANELLISTS' COMMENTS

- I think this capability is important to maintain the ongoing relevance of the software and ensure it continues to meet the business needs. Business ownership for the application and decision/approval rights for major changes should be captured in the configuration DB. At minimum (and for SSEs), the configuration DB must include the application ID, name, business application owner, technical SME, description of the application, service recovery document, etc. All the above information required to support the application in production should be a deliverable of the project and part of an implementation checklist.

5.4.1 Identify configuration items

- Full coverage isn't possible

5.4.2 Establish configuration management system

- Small organisations do not have the resources to invest in a CMS.

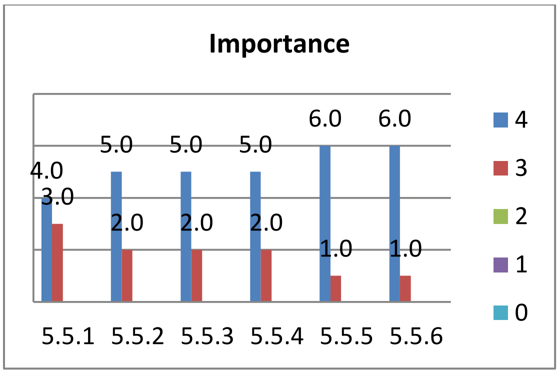
5.4.3 Control configuration items

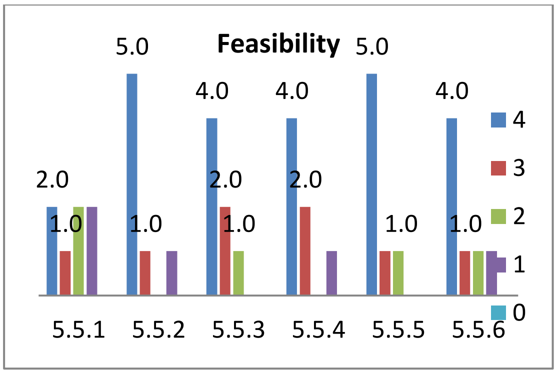
- Not feasible to control fully without a CM

5.4.4 Perform configuration audits

- Time at hand is spent on new projects/issues rather than configuration audits which become a lower priority

5.5 PROBLEM MANAGEMENT





5.5.1 Proactive problem identification

- It would be good for SSEs to proactively identify problems but I'm not sure it would be feasible.
- Staff is usually busy creating/fixing other things rather than proactive identification

5.5.2 Problem categorization and prioritization

- SSEs may not need to proactively identify problems, but they certainly do need to do these following practices.

5.5.3 Problem diagnosis and resolution

- Resolution depends on the feasibility of fixing and the size of task at hand.

5.5.4 Problem closure and evaluation

- Problems closures are often not evaluated.

5.5.5 Major problem review (DC-Sense)

- Only major problem will get reviewed

5.5.6 Problem management reporting

- Not so important for SSEs
- For any major problem, management and reporting will happen. It is generally ignored for smaller problems

[End of Feedback Sheet]

11.7 Second Round Questionnaire

Second Round Questionnaire

Project Title: Holistic Approach to Software Process Improvement in Small Software Enterprises: A Delphi Study

Student Name: Fatina Aweidah
(Masters student, Auckland University of Technology, Faculty of Business and Law, Department of Business Information Systems)

Please attempt the survey with reference to the attached feedback sheet. The feedback sheet provides panellists' ratings and comments in the first round questionnaire. Use the 0-4 scale as per the description table provided below. You also need to categorise the sub processes into basic and advance. Following are the definitions of the scale and categories:

(4) Very Highly Important	Highest relevance to the model Highest impact on the organisation, First priority.	(4) Very Highly Feasible	Can definitely be implemented No increase in available resources needed, Acceptable to all in the organisation.
(3) Very Important	High relevance to the model, High measurable impact on organisation,. Second priority.	(3) Highly Feasible	Can be implemented No major increase in available resources needed, Acceptable to majority in the organisation.
(2) Important	Relevant to the model Has a measurable impact on the Organisation, Low priority.	(2) Feasible	Can be implemented, Slight increase in available resources needed, Acceptable to majority in the organisation.
(1)Low importance	Very low relevance to the model Very low measurable impact on the organisation, No priority.	(1) Not Very Feasible	Some indication that it cannot be implemented, Major increase in available resources needed, Unacceptable to majority in the organisation.
(0) not needed, remove	The practice has no relevance and can be removed from the model.	(0)Not Feasible (Remove)	Cannot be implemented, Extremely large scale increase in available resources needed, Completely unacceptable to all in the organisation.
Basic Process This is a fundamental practice/s for the organisation that lay down the foundation for advanced processes.		**Advanced Process** This is a practice that helps the organisation to improve its quality and maturity level and can be implemented once the organisation has achieved confidence with basic processes.	

Definition of Dynamic Capability

The concept of dynamic capabilities essentially is to ensure enterprise agility and survival as markets and technologies change. There are four types of DC activities:

1. **Sense** to identify opportunities and threats by exploring technological opportunities, scanning and evaluating the markets, listening to customers, along with scanning the other elements of the business environment and quickly carrying out internal and external transformation ahead of the competition.
2. **Seize** by mobilization of resources to address the opportunities.
3. **Transform**: to maintain competitiveness through continual improvement.
4. **Enabler**: is an importance capability/process to establish dynamic capabilities.

1. PROCESS DEFINITION AND IMPROVEMENT (PD&I)

1.1. PROCESS ESTABLISHMENT (DC Enabler)

		Importance	Feasibility	Basic/ Advanced
1.1.1. Initiate process establishment	A business process is a set of activities that takes one or more inputs and creates an output that is of value. • Assign a group to facilitate and manage process establishment. Assign budget and resources			
1.1.2. Define a process	• Define and maintain a description of each process, which shall include : ○ process objectives, owner and target SLAs. ○ Identify the input/s of the process. An input represents the information necessary for a task to begin. Information that is only helpful is not considered an input. ○ determine the set of tasks for the process (WP) ○ Identify the outputs of the process. An output is the result of the work • 1-2 pages of process definition.			
1.1.3. Define interfaces	• Identify all the relationships (interfaces) with other processes or sub-processes. • Identify the target processes that will receive the WP as input. • Identify the requirements to be catered for.			
1.1.4. Define control mechanism	• Identify roles and responsibilities (who is performing the activities).			
1.1.5. Define process measurement	• Define a set of process measures. • Define the parameters that will be used to measure process performance. • Specify procedures for collecting, updating, and retrieving measures. • Light weight measurements that can be handled by SSEs			

Motivation for DC Enabler
Having an established process will enable organisations to measure and improve their processes and customise them to meet arising business needs and changing market conditions. New strategic assets such as building the five capabilities, technology, and customer feedback have to be integrated within the company.

1.2. PROCESS REVIEW

		Importance	Feasibility	Basic/ Advanced
1.2.6. Plan process review	• Develop process assessment plan. • Assign budge, timeline and resources.			
1.2.7. Collect and analyse process data	• Carry out the review to collect process data and measurements.			
1.2.8. Review process performance (DC Sense)	• Analyse process data to understand relative strengths and weaknesses of the process. ○ Key strengths in the process are identified and defined. ○ Key defects in the process are identified and defined.			
1.2.9. Document review results (DC Seize)	• Maintain the results of the review using a standard format in an accessible location.			
1.2.10. Report review result	• Communicate the results of the reviews and the needed action plans to the stakeholders.			

Motivation for DC Sense
Allows the enterprise to measure process performance and assess its alignment with enterprise strategic direction and quality practices.

1.3. PROCESS IMPROVEMENT

		Importance	Feasibility	Basic/ Advanced
1.3.6. Identify and prioritise process improvements	• Identify organisation objectives and prioritise improvements. Organisation commits to the improvements.			
1.3.7. Plan Improvements (DC Seize)	• Develop process improvement plan. ○ Identify the process that needs improvement. ○ Assign budget, timeline, resources, identify scope. (PLAN)			
1.3.8. Develop improvement action plans [New Practice] (DC Seize)	• Analyse the gap between the 'as is' and 'to be' process. • Conduct a root-cause analysis to define the possible reasons for the performance gap. • Develop improvement action plans. • Identify stakeholders who would be impacted by the improvements.			

		Importance	Feasibility	Basic/ Advanced
1.3.9. Implement action plans (DC Transform)	• Implement improvement action plans identified in the review process. Recommended to start on a small group before full implementation.			
1.3.10. Monitor and evaluate improvements (DC Sense, Seize)	• Monitor process implementation and verify against improvement goals. (Check)			
1.3.11. Update organisation process asset library.	• Document the revised process. • Derive lessons learned from defining and implementing organizational process assets. • Make lessons learned available to relevant parties. (ACT)			

Motivation for (DC Seize and Transform)
Allow the enterprise to plan and apply the recommended modifications as well as monitoring process performance for achieving continuous improvements.

Comments:

2. ORGANISATIONAL (ORG)

2.1. BUSINESS ALIGNMENT

		Importance	Feasibility	Basic/ Advanced
2.1.1. Develop strategic business objectives (DC Sense)	• Assessment of external environments. • Assessment of internal strength and weaknesses. Assessment of opportunities and threats. • Set long-term goals and objectives. • Analyse and formulate strategic objectives.			
2.1.2. Identify business outcomes to achieve strategic objectives (DC Seize)	• Identify desired business outcomes. • Prioritisation of these.			
2.1.3. Implement strategic plans. (DC Transform)	• Authorise new investments. • Assess whether strategic objectives are achieved.			

Motivation for (DC Sense, Seize and Transform)
Allows the enterprise to achieve best strategic fit by probing external environment and identifying new business opportunities and threats.

2.2. CUSTOMER RELATIONSHIP

2.2.1. Maintain customer relationships (DC Sense)	• Understand the needs of existing customers and establishes relationships with potential new customers.			
2.2.2. Sign up contracts (DC Seize, Transform)	• Capture customer requirement. • Agree on services and service level targets.			
2.2.3. Carry out customer satisfaction survey	• Plan, carry out customer satisfaction surveys.			
2.2.4. Analyse surveys and develop action plans (DC Sense, Seize).	• Analyse customer surveys. This process provides insights into breached service agreements.			

Motivation for (DC Sense, Seize and Transform)
Sense through feedback from existing relationships and by establishing new relationships. Seize and Transform by maintain the relationship through

assessment of breaches to SLAs and improving the service.

2.3. FINANCIAL PLANNING (DC Enabler)

		Importance	Feasibility	Basic/ Advanced
2.3.1. Develop IT budget	• Develop IT Budget for the next financial period. Allocate the required financial resource for optimum benefits.			
2.3.2. Perform financial analysis and reporting **(DC Sense)**	• Analyse direct and indirect costs of product development and providing services. This process provides insights into the underlying costs, and profitability of the business.			
2.3.3. Establish Invoicing system	• Establish an accurate and clear system of invoicing customers for products and services.			

2.4. HUMAN RESOURCE PLANNING

2.4.1. Identify needed skills and competencies (DC Sense).	• Identify and evaluate the needed skills for the organisation to achieve its goals.			
2.4.2. Recruit qualified staff (DC Seize, Transform)	• Define criteria to evaluate and recruit qualified staff.			
2.4.3. Evaluate and develop staff skills	• Evaluate staff performance and provide feedback. • Define opportunities for staff development.			
2.4.4. Plan training needs (DC Seize)	• Analyse the organization’s business objectives and process improvement plan to identify training needs. • Develop a plan to deliver training. Select people to be trained. Determine cost and resources.			
2.4.5. Deliver and maintain staff training (DC Transform)	• Outsource or deliver training internally; document staff training; assess training effectiveness.			

Motivation for (DC Sense, Seize and Transform)
Securing access to capital and the necessary human resources with the required skills.

3. TECHNICAL SOLUTION DEVELOPMENT (TSD)

3.1. SOFTWARE REQUIREMENTS

		Importance	Feasibility	Basic/ Advanced
3.1.1. Obtain software scope	• This is a requirements backlog. Can be an input from Scope Management (project charter), identifies the high-level software objectives as approved by the customer. OR a new requirement requested by the customer.			
3.1.2. Elicit requirements	• Identify fact-finding sources and techniques to elicit requirements (one-on-one interviews, focus groups, JAD sessions and online surveys). *Requires the development of a profile for each stakeholder (stakeholder register) and stakeholder management strategy.* • *Develop* ***use cases:*** aims at the business audience, customers and users.			
3.1.3. Develop requirements specification	• Analyse the needs of relevant stakeholders • Identify detailed functional and non-functional requirements. • ***Develop requirements specification document***.			
3.1.4. Validate and prioritise requirements	• Requirements review sessions with stakeholders. This validation ensures that requirements accurately reflect users' needs. (correct, unambiguous and relevant). • Trace requirements (previously 3.1.4): requirements must be compared for traceability and consistency with users' requirements (traceable and consistent). *A requirement must be traceable to some source such as to an authoritative source, whether a person or document. Each requirement should have a unique identifier allowing the software design, code, and test procedures to be precisely traced back to the requirement.* • Stakeholders must assign a priority to each requirement (e.g. mandatory, critical, desirable, or optional).			
3.1.5. Establish requirements baseline	• Users agree that requirements are correct and complete. The result is a requirements baseline approved and accepted by the customer. This baseline			

		Importance	Feasibility	Basic/ Advanced
[Changed from 'Develop Requirements baseline']	will be an input to the Requirement Management Process (complete and correct).			
3.1.6. Develop testing strategy [New Practice]	• Identifies high-level business risks of the project that will affect the overall testing strategy; describes the testing approach including how to conduct unit testing, integration testing, system testing, and acceptance testing; describes the testing environment.			
3.1.7. Develop Test Cases	• Develop test cases to verify compliance with requirements (testable).			

3.2. SOFTWARE DESIGN

3.2.1. develop software architecture	• Define the system's architecture (hardware, software and manual operations) • Allocate requirements: every requirement must be allocated to at least one element of the technical architecture. • Define internal and external interfaces: requirements must clearly capture all the interactions with external systems and the external environment so that boundaries are clear. • ***Produce the architectural design document.***			
3.2.2. Verify design	• Ensure backward consistency with the requirements documents.			
3.2.3. Update test cases	• Update test cases to verify compliance with the architectural design.			

3.3. SOFTWARE CONSTRUCTION & TESTING [Changed from Software Construction]

3.3.1 Develop coding standards	• Standards for naming conventions			
3.3.2 Obtain complete test cases	The implementation of this process may differ depending on the lifecycle model used. • A set of tests, test cases (inputs, outputs and test criteria)			
3.3.3 Code software	• Produce the software units defined by the design.			
3.3.4 Verify individual software units	• Verify the individual software units against their developed test cases. [Unit Testing]			
3.3.5 Verify the integrated units	• Combine the software units producing the integrated software product and			

	verify the integrated product against their test cases. [Integration Testing]			
3.3.6 Validate the system [Moved up from 3.4.2]	• Run the system in actual environment or using a simulated test environment and provide users access to assess the system functionality. [User Acceptance Testing]			

3.4. ADDITIONAL SOFTWARE TESTING [Changed from Software Testing]

		Importance	Feasibility	Basic/ Advanced
3.4.1 Apply testing [Changed from 'Apply System Performance testing]	• Compare the integrated system with the non-functional requirements. This may include security, reliability, accuracy, speed and usability. • May include other types of testing: volume, load balance, performance, data conversion and data model testing.			
3.4.2 Perform regression testing [Changed from 3.4.3]	• The need for this process depends on the quality requirements set by the customer. Apply regression testing on modified requirements, design and code.			

Comments:

4. PROJECT MANAGEMENT CAPABILITY

4.1 PROJECT INITIATION

		Importance	Feasibility	Basic/ Advanced
4.1.1. Develop business case (DC Sense, Seize)	This process provides project justifications. • project objectives • analysis of options (comparison of proposal vs. current situation vs. other available options). • critical assumptions and constraints • project and product deliverables • budget and schedule estimates • financial analysis • Identification of 3-4 most critical risks.			
4.1.2. Develop project charter (DC Transform)	Project charter is the contract signed by sponsor and project manager. • develop high level scope, time and budget • measurable objectives and success criteria • project description and boundaries • major product deliverables • summary budget • project success criteria • high level project risks			
4.1.3. Develop stakeholder register and management strategy	• identify primary stakeholders, their roles and interest in the project • consider stakeholder management strategies that will help increase their support throughout the project. • identify stakeholder information needs.			
4.1.4. Develop supplier proposals (DC Sense, Seize, Transform)	• identify potential aspects of product that can be acquired by third party supplier. • identify potential suppliers • obtain proposals • sign supplier contract			

Motivation for 4.1.1 (DC Sense)
This process allows the identification of project constraints, financial, economic and technical feasibility, in addition to project risks.

Motivation for 4.1.2 (DC Transform)
The project charter transforms the presented opportunity into an approved project, once the sponsor signs the contract with the agreed on high level requirements.

Motivation for 4.1.4 (DC Sense, Seize, Transform)
The identification of new or potential suppliers and the negotiation of contracts terms and the sign-off of contracts.

4.2 PROJECT PLANNING

		Importance	Feasibility	Basic/ Advanced

4.2.1. Develop scope	• Collect project requirements
	• **Develop Scope Statement**
	  ○ Product deliverables
	  ○ Project deliverables
	  ○ Technical requirements
	  ○ Limits and exclusions (out of scope product and project deliverables)
	• **Develop WBS**: This is a very essential activity of the planning process, providing levels 1-4 or more. The levels of WBS structure can represent information for different types of stakeholders. For example, level 1 information represents the main project deliverables and is useful to top management; levels 2, 3 are suitable for middle management; and level 4 can be the work package which is the list of activities that will be performed to produce the desired outcomes.			

4.2.2. Develop schedule	• define activity list (The lowest level of the WBS is called a work package. Work packages are activities that have a definite start and stop point, consume resources and represent cost.) • create dependencies between activities • identify resources (HR and other) to complete a work package.			

		Importance	Feasibility	Basic/ Advanced
	• identify time to complete the work packages • produce milestone schedule, which are monitoring points for measuring progress (can be done using a PM software) • **produce Network Diagram** (can be done using a PM software)			
4.2.3. Develop responsibility matrix	• Based on the developed WBS, a responsibility matrix shows the allocation of team members and their roles for each activity. Roles can be for example primary, support and reviewer.			
4.2.4. Develop risk management Plan (DC seize)	• identify major risks (scope, schedule, cost and others) • assess risks in terms of impact, likelihood of occurring • develop strategy to reduce or avoid risk • develop contingency plans • update risk register			
4.2.5. Develop budget	• develop cost estimates of each activity within a work package in the WBS • establish the contingency reserves (cost of risk responses) and management reserves (these may done with the sponsor at the beginning of the project) • **develop schedule baseline.** The contingency reserves are added to the cost estimates to get the cost baseline (management reserves are not included in this cost baseline). • **develop time-phased project budget**. The management reserves plus the cost baseline give the project budget.			
4.2.6. Develop project communication plan	A process is needed for monitoring and controlling the project. • Identify information needed for project stakeholders including sponsors and team members • plan project meetings and reporting to primary stakeholders (scope, schedule, cost and risks) • plan team progress review meetings and reporting • plan team status reporting • plan supplier status reporting			

		Importance	Feasibility	Basic/ Advanced
4.2.7. Plan supplier management plan (DC Seize)	This is an optional process for those organisations that require it. • develop procurement management plan • develop statement of work • plan supplier review meetings to evaluate and quantify risks			

Motivation for 4.2.4, 4.2.7 (DC Seize)
The development of action plans to mitigate the effect of negative risks and or the exploitation and enhancement of of positive risks.

4.3 PROJECT EXECUTION

		Importance	Feasibility	Basic/ Advanced
4.3.1. Manage team	• acquire team (project level) • conduct team training			
4.3.2. Conduct progress review meetings (DC Sense, Seize)	• collect project status information (product performance, risks and quality) ○ conduct review meetings with primary stakeholders ○ conduct team progress review meetings ○ collect supplier status reports ○ conduct risk review meetings (can be part of team review meetings)			
4.3.3. Update project tracking sheets	• implement team access rights to needed information • update schedule tracking sheets • update cost tracking sheets			
4.3.4. Perform supplier management (DC Sense, Seize)	• conduct supplier review meetings • collect supplier deliverables			

Motivation for 4.3.2 and 4.3.4 (DC Sense, Seize)
The carrying out of review meetings reveals project related issues and risks and allows the agreement on planning of corrective actions. It can serve as valuable documentation of a project's life, which can be used to improve the management of future projects.

4.4 PROJECT MONITORING & CONTROLLING (DC Sense, seize, Transform)

4.4.1. Assess scope changes	This process is needed to control scope changes and stop scope creep. • conduct scope review meeting ○ assess change requests ○ approve/ reject project changes.			
4.4.2. Monitor and control schedule and cost	• monitor deviations from baselines. • The cost and schedule baselines are the performance baselines against which the actual performance of the project is measured when the actual activities are performed and costs of the project at any given time are known. • assess the impact of deviations on project • implement corrective actions			
4.4.3. Monitor and control team	• analyse team performance reports • assess the impact of poor performance on project • recommend and implement corrective actions			
4.4.4. Monitor and control risks	• analyse risk reports • assess the impact of risks on project • recommend and implement corrective actions			
4.4.5. Monitor and control supplier	• audit supplier deliverables • accept/reject supplier deliverables.			

Motivation for 4.3.2 and 4.3.4 (DC Sense, Seize)
Project monitoring provides the opportunity to identify variances at early stage so that corrective action can be taken to meet overall project objectives.

4.5 PROJECT CLOSING

		Importance	Feasibility	Basic/ Advanced
4.5.1 run project closure meeting(s)	• This is a meeting with customer and sponsor and other primary stakeholders. *The purpose of the meeting is to review the final acceptance documents and obtain final approval that the project is complete and objectives have been attained* • Stakeholders instruct the closure of the project			

4.5.2	Close supplier contracts	• supplier agreements have been verified and completed			
4.5.3	Close project (DC Sense, Seize)	• Develop a list of outstanding project items. • Work with accounting to ensure all commitments are finalized and closed. • Complete reconciliation of the project budget. • Hold final project review meeting to complete lessons learned report.			
4.5.4	Plan project release [Moved from Support capability]	• Develop a release schedule.			
4.5.5	Deploy the release [Moved from Support capability]	• Establish the release infrastructure on which the software will run. • Develop the infrastructure . • *Release infrastructure covers the hardware, storage, network connections, bandwidth, software licenses, user profiles and access permissions.* • Deploy and test the release..			

Motivation for (DC Sense)
Project closure audit is an important tool to uncover issues, concerns and challenges encountered during the project lifecycle. Along with building an efficient system for capturing lessons learned, which can be used to improve project management processes and to guide future projects.

Comments

5. SUPPORT CAPABILITY

5.1. SERVICE LEVEL MANAGEMENT

		Importa nce	Feasibility	Basic/ Advanced
5.3.1. Identification of service requirements	This process is needed most when the organisation delivers paid services to customers. It is also required when supporting internal departments (SLA and OLA) • To capture desired outcomes (requirements from the customer viewpoint) for new services or major service modifications			
5.3.2. Agreements sign-Off	• Identify service level targets for customers and sign agreements.			
5.3.3. Service level monitoring and reporting	• To monitor achieved service levels and compare them with agreed service level targets. This information is the basis for measures to improve service quality.			

5.2. PROBLEM MANAGEMENT

5.2.1. Proactive problem identification	This is a process is to ensure continuous update of releases: • This process has been added to ensure proactive problem management such as improvement of unreported bugs and security patches.			
5.2.2. Problem categorization and prioritization	• To allow prioritisation of problems			
5.2.3. Problem diagnosis and resolution	• Identify the underlying root cause of a Problem and initiate the most appropriate and economical problem solution.			
5.2.4. Problem closure and evaluation **(DC- Seize)**	• To ensure that, after a successful problem solution, the problem is documented or existing documentation is updated.			
5.2.5. Major problem review 5.2.6. **(DC- Sense)**	• Review the history of major problems to learn lessons for the future. • Perform trend analysis of problem and its causes.			
5.2.7. Problem management reporting	• To ensure that the customers, related parties and service level management are informed of resolved as well as outstanding problems and their progress.			

Motivation for 5.24 (DC Seize)
This process allows the recording of historic data and lessons learns, which become an input or guidelines for resolving future similar problems, hence improving the service.

Motivation for 5.25 (DC Sense)
Trend analysis is the key to identify the different categories of problems and their root causes. It helps to ensure a proactive approach when problems are detected earlier and dealt with immediately rather than resolving the problem at a later stage. This process provides inputs to improve the infrastructure, IT services and organisational processes.

5.3. CONFIGURATION MANAGEMENT (DC Enabler)

5.3.1 Identify configuration items	• Identify configuration items and baselines. Defines who is authorized to make certain changes to the CMS			
5.3.2 Establish configuration management system	• Set up a change control board whose primary function is to approve or reject all change requests that are set against any baseline			
5.3.3 Control configuration items	• Any modifications to the configuration items are approved and are adequately recorded in the configuration management database.			
5.3.3 Perform configuration audits	• Process Objective: To perform regular checks, ensuring that the information contained in the CMS conforms to what is actually installed in the live production environment.			

Motivation for (DC Enabler)
Depending on the scope of configuration management database, this process is considered an enabler to DCs when the organisation decides to document the lessons learned derived from the review of major problems and changes in the CM database.

5.4. CHANGE MANAGEMENT

5.4.1 Review of request for change	• A change proposal describes a proposed major change. The purpose of change proposals is to communicate a proposed major change and assess its risk, impact and feasibility.			
5.4.2 Assessment of change request	• Assessment of change proposals typically submitted for significant changes.			
5.4.3 Change Planning (Transform)	• To authorize detailed software development project planning, and to assess the resulting project plan prior to authorizing the implementation of change.			
5.4.4 Change deployment authorisation	• To assess if all required change components have been implemented and properly tested, and to authorize the change deployment phase.			
5.4.5 Post Implementation Review and change closure (DC Sense, Seize)	To assess the closure of the change implementation and the achieved results and to make sure mistakes are analysed and lessons learned documented before closure.			

Motivation for 5.4.5 (DC Sense, Seize)
Post implementation review of major changes and failed changes especially when breaches to contracts are involved. Root cause analysis to identify mistakes and lessons learned are documented to prevent future issues on similar future changes.

Appendix D - Quantitative Statistical Results

12.1 Process Development and Improvement (PD&I) - Quantitative Results Summary

1.1 PROCESS ESTABLISHMENT

Round 1	Importance MED	MOD	IQR	CV	%(4)	%(0)	Feasibility MED	MOD	IQR	CV	%(4)	%(0)
1.1.1 Initiate process establishment	4	4	2	0.4	56%	0%	2	4	2	0.6	44%	11%
1.1.2 define a process	3	4	2	0.4	33%	0%	3	4	2	0.4	33%	0%
1.1.3 Identify work products	4	4	2	0.4	56%	0%	4	4	1	0.3	56%	0%
1.1.4 Define architecture	3	4	2	0.4	33%	0%	3	4	2	0.4	33%	0%
1.1.5 Define aontrol mechanism	3	3	1	0.3	33%	0%	3	4	2	0.5	33%	11%
1.1.6 Define process measurements	3	2	1	0.4	22%	0%	2	2	1	0.6	22%	11%
1.1.7 document a process	3	4	2	0.4	44%	0%	2	4	2	0.5	33%	0%

Round 2	MED	MOD	IQR	CV	%(4)	%(0)	MED	MOD	IQR	CV	%(4)	%(0)	B(%)	A(%)	MOD	IQR	level
1.1.1 Initiate process establishment	4	4	0.5	0.2	71%	0%	3	3	1	0.2	43%	0%	86%	14%	1	0	B
1.1.2 Define a process	3	3	1	0.2	43%	0%	3	3	0.5	0.2	14%	0%	86%	14%	1	0	B
1.1.3 Define interfaces	3	3	0.5	0.4	14%	0%	3	3	1	0.3	14%	0%	57%	43%	1	1	B
1.1.4 Define Ccontrol mechanism	4	4	1	0.1	57%	0%	3	3	0.5	0.1	29%	0%	86%	14%	1	0	B
1.1.5 Define process measurements	3	3	1	0.2	43%	0%	2	3	1.5	0.3	29%	0%	71%	29%	2	0.5	B

1.2 PROCESS REVIEW

Round 1	MED	MOD	IQR	CV	%(4)	%(0)	MED	MOD	IQR	CV	%(4)	%(0)
1.2.1 Plan process review	3	3	0.5	0.4	13%	0%	2	2	1.3	0.5	13%	0%
1.2.2 Collect and analyse process data	3	3	0.8	0.4	25%	0%	1	1	2.3	0.6	25%	0%
1.2.3 Review process performance	3	4	1.5	0.4	38%	0%	2	3	1.3	0.4	13%	0%
1.2.4 Document review results	3.5	4	1.5	0.4	50%	0%	2	2	1.5	0.5	25%	0%
1.2.5 Report Review Results	3	4	1.5	0.4	38%	0%	3	3	1.5	0.5	25%	0%

Round 2	MED	MOD	IQR	CV	%(4)	%(0)	MED	MOD	IQR	CV	%(4)	%(0)	B(%)	A(%)	MOD	IQR	Level
1.2.1 Plan process review	3	3	1	0.2	43%	0%	3	4	1.5	0.3	43%	0%	43%	57%	1	1	A
1.2.2 Collect and analyse process data	3	3	1	0.3	29%	0%	2.5	2	1.5	0.3	29%	0%	43%	57%	2	1	A
1.2.3 Review process performance	3	3	1	0.2	43%	0%	2.5	2	1.5	0.3	29%	0%	43%	57%	2	1	A
1.2.4 Document review results	3	3	1	0.4	14%	0%	2.5	3	1	0.3	14%	0%	43%	57%	1	1	A
1.2.5 Report review results	3	3	1	0.2	43%	0%	2.5	2	1.5	0.3	29%	0%	43%	57%	1	1	A

1.3 PROCESS IMPROVEMENT

Round 1	MED	MOD	IQR	CV	%(4)	%(0)	MED	MOD	IQR	CV	%(4)	%(0)
1.3.1 Identify and prioritise process improvements	4	4	1.0	0.4	56%	0%	3	4	2	0.4	44%	0%
1.3.2 Plan improvements	3	4	1.0	0.4	44%	0%	3	3	1	0.4	11%	0%
1.3.3 Implement process action plans	4	4	1.0	0.4	56%	0%	3	3	1	0.4	22%	0%
1.3.4 Monitor and evaluate improvements	3	4	1.0	0.4	44%	0%	2	2	1	0.4	11%	0%
1.3.5 Update organisation process asset library	3	4	1	0.4	44%	0%	2	2	1	0.5	22%	0%

Round 2	MED	MOD	IQR	CV	%(4)	%(0)	MED	MOD	IQR	CV	%(4)	%(0)	B(%)	A(%)	MOD	IQR	
1.3.1 Identify and prioritise Process Improvements	3	3	1	0.3	29%	0%	3	3	1.5	0.501	22%	11%	43%	57%	2	1	A
1.3.2 Plan improvements	3	4	1.5	0.3	43%	0%	3	2	1.5	0.315	29%	0%	43%	57%	1	1	A
1.3.3 Develop improvement action plans	3	3	1	0.2	0%	0%	3	3	1	0.38	14%	0%	43%	57%	1	1	A
1.3.4 Implement process action plans	3	3	1	0.3	29%	0%	3	3	1	0.38	14%	0%	43%	57%	1	1	A
1.3.5 Monitor and evaluate improvements	3	3	1	0.3	29%	0%	2	2	1	0.402	14%	0%	43%	57%	2	1	A
1.3.6 Update organisation's process asset library.	2	2	1	0.3	0%	0%	2	2	1	0.402	14%	0%	57%	43%	1	1	A

MED: Median CV: Coefficient of variation %(4): Rating 4,very high
MOD: Mode IQR: Interquantile Range %(0): Rating 0,reomove

12.2 **Organisational (ORG) - Quantitative Results Summary**

2.1 BUSINESS ALIGNMENT

	Importnace						Feasibility					
Round 1	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
2.1.1 Identify vision and goals, develop business objectives	4	4	0	0.1	78%	0%	4	4	1	0.3	56%	0%
2.1.2 Identify business outcomes to achieve strategic objectives	4	4	1	0.1	67%	0%	3	4	1	0.3	44%	0%
2.1.3 Achieve strategic IT plans.	4	4	1	0.2	56%	0%	3	4	2	0.3	33%	0%

	Importnace						Feasibility										
Round 2	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
2.1.1 Develop strategic business objectives	4	4	0	0.2	86%	0%	4	4	1.5	0.3	57%	0%	71%	29%	1	0.5	B
2.1.2 Identify business outcomes to achieve strategic objectives	4	4	0	0.2	86%	0%	3	4	1.5	0.4	43%	0%	71%	29%	1	0.5	B
2.1.3 Implement strategic plan	4	4	0	0.2	86%	0%	4	4	1.5	0.4	57%	0%	57%	43%	1	1.0	B

2.2 CUSTOMER RELATIONSHIP

	Importnace						Feasibility					
Round 1	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
2.2.1 Maintain Customer Relationships	4	4	0	0.0	100%	0%	4	4	1	0.2	56%	0%
2.2.2 Sign up Contracts	4	4	0	0.0	100%	0%	4	4	1	0.4	56%	0%
2.2.3 Customer Satisfaction Survey	4	4	0	0.1	89%	0%	3	4	1	0.2	44%	0%
2.2.4 Analyse Customers' surveys and develop action plans	4	4	1	0.3	67%	0%	4	4	1	0.2	56%	0%

	Importnace						Importnace										
Round 2	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
2.2.1 Maintain Customer Relationships	4	4	0.5	0.1	71%	0%	4	4	0	0.1	86%	0%	100%	0%	1	0	B
2.2.2 Sign up Contracts	4	4	0.5	0.1	71%	0%	4	4	0.5	0.1	71%	0%	100%	0%	1	0	B
2.2.3 Carry out Customer Satisfaction Survey	4	4	0.5	0.1	71%	0%	4	4	1	0.1	57%	0%	71%	29%	1	0.8	B
2.2.4 Analyse surveys and develop action plans.	4	4	0.5	0.1	71%	0%	4	4	0.5	0.1	71%	0%	71%	29%	1	0.8	B

2.3 FINANCIAL PLANNING

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
2.3.1 Financial Planning	4	4	2	0.4	56%	0%	3	3	0	0.3	22%	0%
2.3.2 Develop IT Budget	4	4	1	0.3	56%	0%	3	3	0	0.3	11%	0%
2.3.3 Financial Analysis and reporting	3	3	1	0.3	33%	0%	3	3	1	0.3	33%	0%
2.3.4 Invoicing	4	4	2	0.3	56%	0%	4	4	1	0.3	56%	0%

Round 2	Importnace						Feasibility										
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
2.3.1 Develop IT Budget	4	4	1.5	0.3	57%	0%	3	4	1.5	0.3	43%	0%	86%	14%	1	0	B
2.3.2 Perform Financial Analysis and Reporting	4	4	0.5	0.1	71%	0%	3	3	1	0.3	43%	0%	86%	14%	1	0	B
2.3.3 Establish Invoicing System	4	4	0.5	0.3	71%	0%	4	4	1	0.3	57%	0%	86%	14%	1	0	B

2.4 HUMAN RESOURCE PLANNING

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
2.4.1 Identify needed skills and competencies.	4	4	1	0.3	56%	0%	3	4	1	0.3	44%	0%
2.4.2 Recruit qualified staff	3	3	1	0.2	44%	0%	3	3	1	0.3	44%	0%
2.4.3 Evaluate and develop staff skills	4	4	1	0.3	56%	0%	3	3	1	0.3	33%	0%
2.4.4 Plan training needs.	4	4	1	0.3	56%	0%	3	3	1	0.4	22%	0%
2.4.5 Deliver and maintain staff training	4	4	1	0.3	56%	0%	3	3	1	0.4	22%	0%

Round 2	Importnace						Feasibility										
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
2.4.1 Identify needed skills and competencies	3	3	0.5	0.1	29%	0%	3	3	0	0.1	14%	0%	100%	0%	1	0	B
2.4.2 Recruit qualified staff	3	3	1	0.2	43%	0%	3	3	0.5	0.3	14%	0%	100%	0%	1	0	B
2.4.3 Evaluate and develop staff skills	3	3	1	0.2	43%	0%	3	3	1	0.4	29%	0%	100%	0%	1	0	B
2.4.4 Plan training needs	4	4	1	0.2	57%	0%	3	3	1	0.4	14%	0%	86%	14%	1	0	B
2.4.5 Deliver and maintain staff training	3	3	1	0.2	43%	0%	3	3	1	0.4	14%	0%	86%	14%	1	0	B

2.5 KNOWLEDGE MANAGEMENT

	Importnace						Feasibility					
Round 1	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
2.5.1 Develop KM infrastructure	4	4	4	0.9	56%	44%	1	0	3	1.1	22%	44%
2.5.2 Establish a network of experts	3	0	3	1.0	22%	44%	0	0	2	1.1	22%	44%
2.5.3 Capture and Disseminate knowledge	0	0	3	1.2	22%	56%	1	0	3	1.3	22%	56%
2.5.4 Maintain knowledge system	0	0	3	1.2	22%	56%	1	0	3	1.2	11%	44%

Round 2

Knowledge management process was dropped in the second round

MED: Median
MOD: Mode
CV: Coefficient of variation
IQR: Interquantile Range
%(4): Rating 4,very high
%(0): Rating 0,reomove

12.3 **Technical Solution Development (TSD) - Quantitative Results Summary**

3.1 SOFTWARE REQUIREMENTS

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
3.1.1 Obtain software scope	**4**	**4**	**0**	**0**	**100%**	**0**	4	4	0	0.2	78%	0%
3.1.2 Elicit requirements	4	4	0	0.1	89%	0%	4	4	0	0.2	78%	0%
3.1.3 Analyse Requirements	4	4	0	0.0	100%	0%	4	4	0	0.2	89%	0%
3.1.4 Trace requirements	4	4	0	0.2	89%	0%	4	4	1	0.3	56%	0%
3.1.5 Validate and prioritise requirements	4	4	0	0.1	89%	0%	4	4	0	0.2	78%	0%
3.1.6 Develop requirements baseline	4	4	0	0.0	100%	0%	4	4	0	0.2	89%	0%
3.1.7 Develop test cases	4	4	0	0.0	100%	0%	4	4	0	0.176	89%	0%

1= Basic 2= Advanced

Round 2	Importnace						Feasibility										
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
3.1.1 Obtain software scope	4	4	0	0.1	86%	0%	4	4	1	0.1	57%	0%	100%	0%	1	0.0	B
3.1.2 Elicit requirements	4	4	0	0.1	86%	0%	3	4	1	0.3	43%	0%	86%	14%	1	0.0	B
3.1.3 Develop requirements specifications	4	4	0	0.1	86%	0%	3	3	1	0.2	43%	0%	100%	0%	1	0.0	B
3.1.4 Validate and prioritise requirements	4	4	0	0.1	86%	0%	3	3	0.5	0.1	29%	0%	57%	43%	1	1.0	B
3.1.5 Establish requirements baseline	4	4	0	0.1	86%	0%	3	3	1	0.2	43%	0%	100%	0%	1	0.0	B
3.1.6 Develop testing strategy	4	4	0	0.1	86%	0%	3	3	0.5	0.2	29%	0%	100%	0%	1	0.0	B
3.1.7 Develop test cases	4	4	0	0.1	86%	0%	3	3	0.5	0.3	29%	0%	100%	0%	1	0.0	B

3.2 SOFTWARE DESIGN

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
3.2.1 Develop software architectural	4	4	0	0.0	100%	0%	4	4	1	0.2	67%	0%
3.2.2 Verify design	4	4	0	0.1	89%	0%	4	4	1	0.2	56%	0%
3.2.3 Update test cases	4	4	0	0.1	89%	0%	4	4	1	0.3	67%	0%

Round 2	Importnace						Feasibility										
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
3.2.1 Develop software architectural	3	3	1	0.2	43%	0%	3	3	1	0.3	14%	0%	43%	57%	2	1.0	A
3.2.2 Verify design	4	4	1	0.2	57%	0%	3	3	1	0.3	14%	0%	29%	71%	2	1.0	A
3.2.3 Update test cases	3	3	1	0.2	43%	0%	3	3	0.5	0.2	0%	0%	43%	57%	2	1.0	A

3.3

Round 1

3.3 SOFTWARE CONSTRUCTION	Importnace MED	MOD	IQR	CV	(4%)	(%0)	Feasibility MED	MOD	IQR	CV	(4%)	(%0)
3.3.1 Develop coding standards	4	4	0	0.1	89%	0%	4	4	1	0.3	67%	0%
3.3.2 Obtain complete test cases	4	4	0	0.1	89%	0%	4	4	1	0.2	67%	0%
3.3.3 Code software product	4	4	0	0.0	100%	0%	4	4	0	0.2	89%	0%
3.3.4 Verify individual software units	4	4	0	0.1	89%	0%	4	4	1	0.3	67%	0%
3.3.5 Verify the integrated units	4	4	0	0.0	100%	0%	4	4	0	0.2	78%	0%

Round 2

3.3 SOFTWARE CONSTRUCTION & Testing	Importnace MED	MOD	IQR	CV	(4%)	(%0)	Feasibility MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
3.3.1 Develop coding standards	4	4	1.5	0.4	57%	0%	2	2	1	0.4	14%	0%	71%	29%	1	0.5	B
3.3.2 Obtain complete test cases	4	4	1.5	0.4	57%	0%	3	2	1.5	0.4	29%	0%	86%	14%	1	0.0	B
3.3.3 Code software product	4	4	1	0.2	57%	0%	3	3	1	0.2	43%	0%	100%	0%	1	0.0	B
3.3.4 Verify individual software units	4	4	0.5	0.2	71%	0%	4	4	0.5	0.2	57%	0%	86%	14%	1	0.0	B
3.3.5 Verify the integrated units	4	4	0.5	0.2	71%	0%	4	4	1	0.2	57%	0%	71%	29%	1	0.5	B
3.3.6 Validate the System	4	4	0.5	0.2	71%	0%	4	4	1	0.2	57%	0%	86%	14%	1	0.0	B

3.4

Round 1

3.4 SOFTWARE TESTING	Importnace MED	MOD	IQR	CV	(4%)	(%0)	Feasibility MED	MOD	IQR	CV	(4%)	(%0)
3.4.1 Apply system performance testing	4	4	0	0.0	100%	0%	4	4	0.5	0.2	71%	0%
3.4.2 Validate the system	4	4	0	0.0	100%	0%	4	4	1	0.3	71%	0%
3.4.3 Perform regression testing	4	4	0	0.0	100%	0%	4	4	1	0.3	71%	0%

Round 2

3.4 ADDITIONAL SOFTWARE TESTING	Importnace MED	MOD	IQR	CV	(4%)	(%0)	Feasibility MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
3.4.1 Apply testing	3	3	1	0.2	43%	0%	3	3	1	0.2	0%	0%	83%	17%	1	0.0	B
3.4.2 Perform regression testing	4	4	1.5	0.4	57%	0%	3	3	1	0.4	29%	0%	100%	0%	1	0.0	B

MED: Median
MOD: Mode
CV: Coefficient of variation
IQR: Interquantile Range
%(4): Rating 4,very high
%(0): Rating 0,reomove

12.4 Project Management (PM) Quantitative Results Summary

4.1 PROJECT INITIATION

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
4.1.1. Perform feasibility analysis	4	4	0	0.2	89%	0%	4	4	2	0.3	67%	0%
4.1.2. Initiate scope	4	4	0	0.2	78%	0%	4	4	1	0.3	56%	0%
4.1.3. Initiate schedule	4	4	0	0.2	78%	0%	4	4	1	0.3	67%	0%
4.1.4. Initiate budget	4	4	1	0.3	67%	0%	4	4	2	0.3	56%	0%
4.1.5. Initiate communication management	4	4	0	0.2	78%	0%	4	4	1	0.3	67%	0%
4.1.6. Initiate team	4	4	0	0.2	78%	0%	4	4	2	0.3	56%	0%
4.1.7. Initiate risk analysis	4	4	0	0.2	78%	0%	4	4	2	0.3	56%	0%
4.1.8.Initiate supplier management	4	4	1	0.2	67%	0%	3	4	2	0.3	44%	0%

1= Basic 2= Advanced

Round 2	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
4.1.1. Develop business case	4	4	0.5	0.2	71%	0%	3	3	1	0.2	43%	0%	57%	43%	1	1.0	B
4.1.2. Develop project charter	4	4	1	0.2	57%	0%	3	3	1	0.2	43%	0%	100%	0%	1	0.0	B
4.1.3. Develop stakeholder register and management strategy	4	4	1	0.2	57%	0%	3	4	2	0.3	43%	0%	57%	43%	1	1.0	B
4.1.4. Develop supplier proposals	4	4	1	0.2	57%	0%	3	4	1	0.2	43%	0%	86%	14%	1	0.0	B

4.2 PROJECT PLANNING

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
4.2.1 Plan scope	4	4	0	0.2	78%	0%	4	4	1	0.3	67%	0%
4.2.2 Plan schedule	4	4	0	0.2	78%	0%	4	4	1	0.3	67%	0%
4.2.3 Plan cost	4	4	0	0.2	78%	0%	4	4	1	0.3	67%	0%
4.2.4 Plan quality	4	4	1	0.3	56%	0%	3	4	2	0.3	44%	0%
4.2.5 Plan communication	4	4	1	0.2	67%	0%	4	4	2	0.3	56%	0%
4.2.6 Plan human resources	4	4	1	0.2	67%	0%	4	4	2	0.4	56%	0%
4.2.7 Plan Risks	4	4	0	0.2	78%	0%	4	4	1	0.3	56%	0%
4.2.8 Plan supplier management	4	4	1	0.3	67%	0%	4	4	2	0.4	56%	0%

Round 2	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
4.2.1 Develop scope	4	4	0.5	0.2	71%	0%	3	3	0.5	0.2	29%	0%	67%	33%	1	0.8	B
4.2.2 Develop schedule	4	4	0.5	0.2	71%	0%	3	3	0.5	0.2	29%	0%	100%	0%	1	0.0	B
4.2.3. Develop responsibility matrix	4	4	1	0.2	57%	0%	3	3	0.5	0.2	29%	0%	100%	0%	1	0.0	B
4.2.4. Develop risk management plan	4	4	0.5	0.2	71%	0%	3	3	0.5	0.2	29%	0%	100%	0%	1	0.0	B
4.2.5. Develop budget	4	4	0.5	0.2	71%	0%	3	3	0.5	0.2	29%	0%	83%	17%	1	0.0	B
4.2.6. Develop project communication plan	4	4	1	0.2	57%	0%	3	3	0.5	0.2	29%	0%	100%	0%	1	0.0	B
4.2.7. Develop supplier management plan	3	4	1.5	0.3	43%	0%	3	3	0.5	0.2	29%	0%	83%	17%	1	0.0	B

4.3 PROJECT EXECUTION

	Importnace						Feasibility					
Round 1	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
4.3.1 Execute scope	4	4	0	0.2	88%	0%	4	4	0	0.3	75%	0%
4.3.2 Execute schedule	4	4	0	0.2	88%	0%	4	4	0	0.3	75%	0%
4.3.3 Execute cost	4	4	0	0.2	88%	0%	4	4	1	0.3	63%	0%
4.3.4 Execute quality management plan	4	4	0	0.2	88%	0%	2	2	2	0.4	50%	0%
4.3.5 Perform communication management	4	4	0	0.2	88%	0%	4	4	2	0.3	50%	0%
4.3.6 Execute human resource management	4	4	0	0.3	75%	0%	3	4	2	0.4	38%	0%
4.3.7 Execute risk management plan	4	4	0	0.2	88%	0%	4	4	1	0.3	50%	0%
4.3.8 Perform supplier management	4	4	0	0.3	75%	0%	3	4	2	0.4	38%	0%

	Importnace						Feasibility						1= Basic 2= Advanced				
Round 2	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
4.3.1. Manage team	3	4	1.5	0.3	43%	0%	3	3	0.5	0.2	29%	0%	86%	0%	1	0.0	B
4.3.2. Conduct progress review meeting	4	4	0.5	0.2	71%	0%	3	4	1.5	0.3	43%	0%	86%	0%	1	0.0	B
4.3.3. Update project tracking sheets	4	4	0.5	0.2	71%	0%	3	3	1	0.3	29%	0%	86%	0%	1	0.0	B
4.3.4. Perform Supplier management	3	3	1	0.3	29%	0%	3	3	0.5	0.2	14%	0%	57%	29%	1	0.8	B

4.4 PROJECT MONITORING & CONTROLLING

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
4.4.1 Integrated change control	4	4	1	0.2	67%	0%	4	4	1	0.3	67%	0%
4.4.2 Monitor and control scope	4	4	0	0.2	78%	0%	4	4	1	0.3	67%	0%
4.4.3 Monitor and control schedule and cost	4	4	0	0.2	78%	0%	4	4	1	0.2	67%	0%
4.4.4 Monitor and control cost	4	4	1	0.3	56%	0%	4	4	1	0.3	56%	0%
4.4.5 Monitor and control quality	4	4	0	0.2	78%	0%	4	4	2	0.3	56%	0%
4.4.6 Monitor and control communication	4	4	0	0.2	78%	0%	3	4	1	0.3	44%	0%
4.4.7 Monitor and control human resources	4	4	1	0.2	67%	0%	3	3	1	0.3	33%	0%
4.4.8 Monitor and control risks	4	4	0	0.2	78%	0%	4	4	1	0.2	56%	0%
4.4.9 Monitor and control supplier	4	4	1	0.2	67%	0%	3	3	1	0.3	33%	0%

Round 2	Importnace						Feasibility										
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
4.4.1. Assess scope changes	4	4	0.5	0.2	71%	0%	3	3	0.5	0.2	29%	0%	100%	0%	1	0.0	B
4.4.2. Monitor and control schedule and cost	4	4	0.5	0.2	71%	0%	4	4	1	0.2	57%	0%	100%	0%	1	0.0	B
4.4.3. Monitor and control Team	3	3	1	0.2	43%	0%	3	3	0.5	0.4	14%	0%	67%	33%	1	0.8	B
4.4.4. Monitor and control risks	4	4	0.5	0.2	71%	0%	3	3	0.5	0.2	14%	0%	67%	33%	1	0.8	B
4.4.5. Monitor and control supplier	3	3	0.5	0.2	29%	0%	3	3	0	0.1	0%	0%	67%	33%	1	0.8	B

4.5 PROJECT CLOSING

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
4.5.1 Close supplier contracts	4	4	0	0.2	88%	0%	4	4	2	0.4	63%	0%
4.5.2 Update project documentation	3.5	4	1.25	0.3	50%	0%	3.5	4	2	0.4	50%	0%

Round 2	Importnace						Feasibility										
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
4.5.1. Run project closure meeting(s)	3	4	1	0.2	43%	0%	3	3	0.5	0.2	14%	0%	83%	17%	1	0.0	B
4.5.2. Close supplier contracts	3	3	1	0.2	43%	0%	3	3	0.5	0.4	14%	0%	100%	0%	1	0.0	B
4.5.3. Close project	4	4	1.5	0.3	57%	0%	3	4	1	0.2	43%	0%	100%	0%	1	0.0	B
4.5.4. Plan project release	4	4	1	0.2	57%	0%	3	3	0	0.2	14%	0%	100%	0%	1	0.0	B
4.5.5. Deploy release	4	4	1	0.2	57%	0%	3	3	0	0.2	14%	0%	100%	0%	1	0.0	B

MED: Median
MOD: Mode

CV: Coefficient of variation
IQR: Interquantile Range

%(4): Rating 4,very high
%(0): Rating 0,reomove

12.5 **Support (SUP) - Quantitative Results Summary**

5.1 SERVICE LEVEL MANAGEMENT

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
5.1.1 Identification of service requirements	4	4	1	0.302	63%	0%	4	4	1.25	0.386	63%	0
5.1.2 Agreements sign-off	4	4	0.25	0.283	75%	0%	4	4	1.25	0.386	63%	0
5.1.3 Service level monitoring and reporting	4	4	1	0.3	63%	0%	3.5	4	1.25	0.394	50%	0%

Round 2	Importnace						Feasibility										
5.1 SERVICE LEVEL MANAGEMENT	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
5.1.1 Identification of service requirements	4	4	0	0.1	86%	0%	3	3	1	0.2	43%	0%	100%	0%	1	0.0	B
5.1.2 Agreements sign-off	4	4	0	0.1	86%	0%	4	4	1	0.1	57%	0%	100%	0%	1	0.0	B
5.1.3 Service level monitoring and reporting	4	4	0	0.1	86%	0%	4	4	1	0.1	57%	0%	71%	29%	1	0.5	B

5.5 PROBLEM MANAGEMENT

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
5.5.1 Proactive problem identification	4	4	1	0.208	57%	0%	2	4	2	0.891	29%	0%
5.5.2 Problem categorization and prioritization	4	4	0.5	0.18	71%	0%	4	4	0.5	0.467	71%	0%
5.5.3 Problem diagnosis andresolution	4	4	0.5	0.18	71%	0%	4	4	1	0.324	57%	0%
5.5.4 Problem closure and evaluation	4	4	0.5	0.18	71%	0%	4	4	1	0.487	57%	0%
5.5.5 Major problem review	4	4	0	0.132	86%	0%	4	4	0.5	0.306	71%	0%
5.5.6 Problem management reporting	4	4	0	0.132	86%	0%	4	4	1.5	0.567	57%	0%

1= Basic 2= Advanced

Round 2	Importnace						Feasibility										
5.2	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
5.2.1 Proactive problem identification	4	4	1	0.2	57%	0%	3	3	1	0.3	14%	0%	86%	14%	1	0.0	B
5.2.2 Problem categorisation and prioritisation	4	4	0.5	0.1	71%	0%	3	2	1.5	0.3	29%	0%	100%	0%	1	0.0	B
5.2.3 Problem diagnosis and resolution	4	4	1	0.1	57%	0%	3	4	1.5	0.3	43%	0%	71%	29%	1	0.5	B
5.2.4 Problem closure and evaluation	3	3	1	0.4	29%	0%	3	3	0.5	0.4	14%	0%	100%	0%	1	0.0	B
5.2.5. Major problem review (DC- Sense)	3	4	1	0.3	43%	0%	3	3	1	0.4	29%	0%	57%	43%	1	1.0	B
5.2.6. Problem management reporting	4	4	1	0.2	57%	0%	3	3	0	0.1	14%	0%	100%	0%	1	0.0	B

5.4 CONFIGURATION MANAGEMENT

Round 1	Importnace MED	MOD	IQR	CV	(4%)	(%0)	Feasibility MED	MOD	IQR	CV	(4%)	(%0)
5.4.1 Identify configuration items	4	4	0	0.279	86%	0%	4	4	2	0.499	57%	0%
5.4.2 Establish a configuration management system	4	4	0	0.279	86%	0%	4	4	2	0.645	57%	0%
5.4.3 Control configuration items	4	4	0	0.279	86%	0%	4	4	2	0.499	57%	0%
5.4.4 Perform configuration audits	4	4	0.5	0.306	71%	0%	3	4	2	0.5	43%	0%

Round 2 5.3	Importnace MED	MOD	IQR	CV	(4%)	(%0)	Feasibility MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
5.3.1 Identify configuration items	4	4	1	0.1	57%	0%	3	4	1.5	0.3	40%	0%	43%	57%	2	1.0	A
5.3.2 Establish a configuration management system	4	4	1	0.1	57%	0%	3	3	0.5	0.2	29%	0%	71%	29%	1	0.8	B
5.3.3 Control configuration items	4	4	1	0.1	57%	0%	3	3	1	0.2	43%	0%	71%	29%	1	0.8	B
5.3.4 Perform configuration audits	3	3	1	0.2	43%	0%	3	3	1	0.3	14%	0%	43%	57%	2	1.0	A

5.2 CHANGE MANAGEMENT

Round 1	Importnace MED	MOD	IQR	CV	(4%)	(%0)	Feasibility MED	MOD	IQR	CV	(4%)	(%0)
5.2.1 Review of request for change	4	4	0.5	0.306	71%	0%	4	4	1	0.402	71%	0%
5.2.2 Assessment of change request	4	4	0.5	0.306	71%	0%	4	4	1	0.402	71%	0%
5.2.3 Change evaluation authorisation	4	4	0.5	0.306	71%	0%	4	4	2	0.499	57%	0%
5.2.4 Change deployment authorisation	4	4	0.5	0.306	71%	0%	4	4	2	0.499	57%	0%
5.2.5 Post implementation review and change closure	4	4	0.5	0.306	71%	0%	4	4	2	0.499	57%	0%

Round 2 5.4	Importnace MED	MOD	IQR	CV	(4%)	(%0)	Feasibility MED	MOD	IQR	CV	(4%)	(%0)	B(%)	A(%)	MOD	IQR	
5.4.1 Review of request for change	4	4	1	0.15	0.571	0	3	3	1	0.156	43%	0%	86%	14%	1	0.0	B
5.4.2 Assessment of change request	4	4	1	0.15	0.571	0	3	3	0.5	0.149	29%	0%	86%	14%	1	0.0	B
5.4.3 Change Planning	4	4	0.75	0.141	0.667	0	3.5	4	1	0.245	50%	0%	67%	33%	1	0.8	B
5.4.4 Change deployment authorisation	4	4	0.5	0.1	71%	0%	3	4	1	0.2	43%	0%	71%	29%	1	0.5	B
5.4.5 Post implementation review and closure	4	4	1.5	0.4	57%	0%	3	3	1	0.4	29%	0%	57%	43%	1	1.0	B

5.3 RELEASE MANAGEMENT

Round 1	Importnace						Feasibility					
	MED	MOD	IQR	CV	(4%)	(%0)	MED	MOD	IQR	CV	(4%)	(%0)
5.3.1 Release planning	4	4	0.5	0.306	71%	0%	4	4	1	0.402	71%	0%
5.3.2 Establish the release infrastructure	4	4	0.5	0.306	71%	0%	4	4	1	0.402	71%	0%
5.3.3 Release documentation	4	4	0.5	0.306	71%	0%	4	4	1.5	0.416	57%	0%
5.3.4 Deploy the release	4	4	0.5	0.306	71%	0%	4	4	1	0.402	71%	0%

Round 2

In the second version of the model, Release Management is moved to 5.4 Project Closing

Appendix E - Final Model Version 2

PROCESS DEVELOPMENT & IMPROVEMENT (PD&I)	Dynamic Capabilities	Classification
1.1 PROCESS ESTABLISHMENT	DC Enabler	
1.1.1 Initiate process establishment		B
1.1.2 Define a process		B
1.1.3 Define interfaces		B
1.1.4 Define control mechanism		B
1.1.5 Define process measurements		B
1.2 PROCESS REVIEW		
1.2.1 Plan process review		A
1.2.2 Collect and analyse process data		A
1.2.3 Review process performance	Sense	A
1.2.4 Document review results	Seize	A
1.2.5 Report review results		A
1.3 PROCESS IMPROVEMENT		
1.3.1 Identify and prioritise Process Improvements		A
1.3.2 Plan improvements	Seize	A
1.3.3 Develop improvement action plans	Seize	A
1.3.4 Implement process action plans	Transform	A
1.3.5 Monitor and evaluate improvements	Sense, Seize	A
1.3.6 Update organisation's process asset library.		A

ORGANISATIONAL (ORG)	Dynamic Capabilities	Classification
2.1 BUSINESS ALIGNMENT		
1.1 Develop strategic business objectives	Sense	B
2.1.2 Identify business outcomes to achieve strategic objectives	Seize	B
2.1.3 Implement strategic plan	Transform	B
2.2 CUSTOMER RELATIONSHIP		
2.2.1 Maintain customer relationships	Sense	B
2.2.2 Sign up contracts	Seize, Transform	B
2.2.3 Carry out customer satisfaction survey		B
2.2.4 Analyse surveys and develop action plans.	Sense, Seize	B
2.3 FINANCIAL PLANNING		
2.3.1 Develop IT budget		B
2.3.2 Perform financial analysis and reporting	Sense	B
2.3.3 Establish invoicing system		B

2.4 HUMAN RESOURCE PLANNING		
2.4.1 Identify needed skills and competencies	Sense	B
2.4.2 Recruit qualified staff	Seize, Transform	B
2.4.3 Evaluate and develop staff skills		B
2.4.4 Plan training needs	Seize	B
2.4.5 Deliver and maintain staff training	Transform	B

TECHNICAL SOLUTION DEVELOPMENT (TSD)	Dynamic Capabilities	Classification
3.1 SOFTWARE REQUIREMENTS		
3.1.1 Obtain software scope		B
3.1.2 Elicit requirements		B
3.1.3 Develop requirements specifications		B
3.1.4 Validate and prioritise requirements		B
3.1.5 Establish requirements baseline		B
3.1.6 Develop testing strategy		B
3.1.7 Develop test cases		B
3.2 SOFTWARE DESIGN		
3.2.1 Develop software architectural		B
3.2.2 Verify design		B
3.2.3 Update test cases		B
3.3 SOFTWARE CONSTRUCTION & TESTING		
3.3.1 Develop coding standards		B
3.3.2 Obtain complete test cases		B
3.3.3 Code software product		B
3.3.4 Verify individual software units		B
3.3.5 Verify the integrated units		B
3.3.6 Validate the System		B
3.4 ADDITIONAL SOFTWARE TESTING		
3.4.1 Apply testing		B
3.4.2 Perform regression testing		B

PROJECT MANAGEMENT (PM)	**Dynamic Capabilities**	**Classifications**
4.1 PROJECT INITIATION		
4.1.1. Develop business case	Sense, Seize	B
4.1.2. Develop project charter	Transform	B
4.1.3. Develop stakeholder register and management strategy		B
4.1.4. Develop supplier proposals	Sense, Seize, Transform	B
4.2 PROJECT PLANNING		
4.2.1 Develop scope		B
4.2.2 Develop schedule		B
4.2.3. Develop responsibility matrix		B
4.2.4. Develop risk management plan	Seize	B
4.2.5. Develop budget		B
4.2.6. Develop project communication plan		B
4.2.7. Develop supplier management plan	Seize	B
4.3 PROJECT EXECUTION		
4.3.1. Manage team		B
4.3.2. Conduct progress review meeting	Sense, Seize	B
4.3.3. Update project tracking sheets		B
4.3.4. Perform Supplier management	Sense, Seize	B
4.4 PROJECT MONITORING & CONTROLLING	**DC Sense, Seize, Transform**	
4.4.1. Assess scope changes		B
4.4.2. Monitor and control schedule and cost		B
4.4.3. Monitor and control Team		B
4.4.4. Monitor and control risks		B
4.4.5. Monitor and control supplier		B
4.5 PROJECT CLOSING		
4.5.1. Run project closure meeting(s)		B
4.5.2. Close supplier contracts		B
4.5.3. Close project	Sense, Seize	B
4.5.4. Plan project release		B
4.5.5. Deploy release		B

SUPPORT (SUP)		
5.1 SERVICE LEVEL MANAGEMENT		
5.1.1 Identification of service requirements		B
5.1.2 Agreements sign-off		B
5.1.3 Service level monitoring and reporting		B
5.2 PROBLEM MANAGEMENT		
5.2.1 Proactive problem identification		B
5.2.2 Problem categorisation and prioritisation		B
5.2.3 Problem diagnosis and resolution		B
5.2.4 Problem closure and evaluation	Seize	B
5.2.5. Major problem review (DC- Sense)	Sense	B
5.2.6. Problem management reporting		B
5.3 CONFIGURATION MANAGEMENT	Dc Enabler	
5.3.1 Identify configuration items		B
5.3.2 Establish a configuration management system		B
5.3.3 Control configuration items		B
5.3.4 Perform configuration audits		B
5.4 CHANGE MANAGEMENT		
5.4.1 Review of request for change		B
5.4.2 Assessment of change request		B
5.4.3 Change Planning		B
5.4.4 Change deployment authorisation		B
5.4.5 Post implementation review and closure	Sense, Seize	B