Investigating the Relationship between Software Process Improvement, Situational Change, and Business Success in Software SMEs

Paul Clarke

B.Sc. (Hons.)

Submitted for the degree of Doctor of Philosophy

School of Computing, Dublin City University

Supervisor: Dr. Rory V. O’Connor

July 2012

Declaration:

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ____________________ ID No.: 92039065

Date: ____________________

Acknowledgements

In life, we all owe a debt of gratitude to certain individuals who have supported our development and advancement. As an infant and young adult, I was most fortunate to have two very generous, loving and supportive parents. And their support has been a constant feature throughout my adult life. To my parents, Paddy and Rose, I therefore offer a heartfelt thank you. On the theme of family, I must also recognise the great patience and flexibility that my wife, Annette, has demonstrated. Throughout all of the late nights in the office and the many surrendered weekends, Annette was always expressly supportive of my efforts. To Annette, I also offer a sincere thank you.

With respect to the specifics of my PhD research, I am most grateful for the support and direction offered by my supervisor, Dr. Rory O'Connor. Rory's commitment, both to research programmes and to his researchers, is highly commendable. He works diligently to maximise the potential of his students and their research, while also supporting student welfare. I'm certain that Rory's efforts are considerably beyond the basic requirements of a Supervisor. I'm equally certain that this thesis has benefited greatly from Rory's attention to detail and quality over the lifetime of my PhD research.

I'm also grateful to a number of the staff in the DCU Business School for their generosity of time in the earlier days of my research. Specifically, Prof. Brian Leavy, Prof. David Jacobson and Dr. Sarah Ingle were instrumental in guiding the business success related aspect of this thesis. On the subject of acknowledgements, it would be remiss of me not to mention those individuals who have peer reviewed various aspects of this work – it is vitally important that a community of independent expert volunteers is accessible in order to develop one's ideas and thinking.

I wish also to thank Lero – the Irish Software Engineering Research Centre – for awarding me the research scholarship that made this thesis possible. And to the many practitioners who participated in the field study: Thank you one and all. Finally, I should mention Murat Yilmaz, a slightly crazy Turk who punctuated my research stress with many bizarre insights. Thank you, Murat.

Contents

List of Tables

List of Figures

Abstract

While we have learned a great deal from Software Process Improvement (SPI) research to date, no earlier study has been designed from the outset to examine the relationship between SPI and business success in software development small- to-medium- sized companies (software SMEs). Since business processes are generally acknowledged as having an important role to play in supporting business success, it follows that the software development process (a large and complex component of the overall business process) has an important contribution to make in supporting business success in software development companies. However, to date we have very little evidence regarding the role of SPI in supporting business success, especially for software SMEs.

The need for SPI is dependent on the extent of situational change in a software development setting, and therefore any examination of the relationship between SPI and business success would be deficient if it did not also examine the extent of situational change. Therefore, this thesis describes a novel approach to examining SPI, situational change and business success in software development companies. Furthermore, having discharged this new approach to 15 software SMEs, this thesis makes the important new discovery that the amount of SPI implemented in a software SME is positively associated with the extent of business success – especially when the degree of situational change is taken into account.

This thesis describes the first published study to examine the relationship between SPI, situational change and business success in software SMEs. The findings suggest that there are business benefits to implementing SPI in software SMEs, with the degree of situational change being an important factor informing SPI initiatives. Furthermore, this research has yielded valuable new insights into the nature of SPI, situational change and business success in software SMEs.

Related Peer Reviewed Publications

Clarke, P., O'Connor, R.V.: A robust approach to examining business success in software development companies. Accepted for publication in the proceedings of the International Conference on Software Quality 2012.

Clarke, P., O'Connor, R.V.: The influence of SPI on business success in software SMEs: An empirical study, Journal of Systems and Software, Vol. 85, Issue 10, pp. 2356-2367, October 2012.

Clarke, P., O'Connor, R.V.: The situational factors that affect the software development process: Towards a comprehensive reference framework, Information and Software Technology, Vol. 54, Issue 5, pp. 433-516, May 2012.

Clarke, P., O'Connor, R.V.: Business success in software SMEs: Recommendations for future SPI studies. To appear in: Proceedings of the 19th European System and Software Process Improvement and Innovation Conference, CCIS 301/2012. Heidelberg, Germany: Springer-Verlag. June 2012.

Clarke, P., O'Connor, R.V.: A hierarchy of SPI activities for software SMEs: results from ISO/IEC 12207-based SPI assessments. To appear in: Proceedings of the 12th International Conference on Software Process Improvement and Capability dEtermination, CCIS 290/2012. Heidelberg, Germany: Springer-Verlag. May 2012.

Clarke, P., O'Connor, R.V.: The Meaning of Success for Software SMEs: An Holistic Scorecard Approach, In: Proceedings of the 18th European System and Software Process Improvement and Innovation Conference, CCIS 172/2011, Heidelberg, Germany: Springer-Verlag. June 2011. *Received Overall Best Research Paper Award*.

Clarke, P., O'Connor, R.V.: An Approach to Evaluating Software Process Adaptation, In: Proceedings of the 11th International Conference on Software Process Improvement and Capability dEtermination, CCIS 155/2011. Heidelberg, Germany: Springer-Verlag. May 2011.

Clarke, P., O'Connor, R.V.: Harnessing ISO/IEC 12207 to Examine the Extent of SPI Activity in an Organisation, In: Proceedings of the 17th European Systems and Software Process Improvement and Innovation Conference, CCIS 99/2010. Heidelberg, Germany: Springer-Verlag. September 2010.

Clarke, P., O'Connor, R.V.: Towards the identification of the influence of SPI on the successful evolution of software SMEs, In: Proceedings of 18th International Software Quality Management Conference. London, U.K.: British Computer Society. March 2010.

Related Publications – Under Peer Review

Clarke, P., O'Connor, R.V.: An empirical examination of the extent of software process improvement in software SMEs. Under review for the Journal of Software Maintenance and Evolution: Research and Practice. Submitted January 2012.

Related Posters

Clarke, P., O'Connor, R.V.: Dynamic Capability in the Software Development Process. 16th annual Software Engineering Process Group (SEPG) Europe Conference. June 2011.

Document Map – Part 1

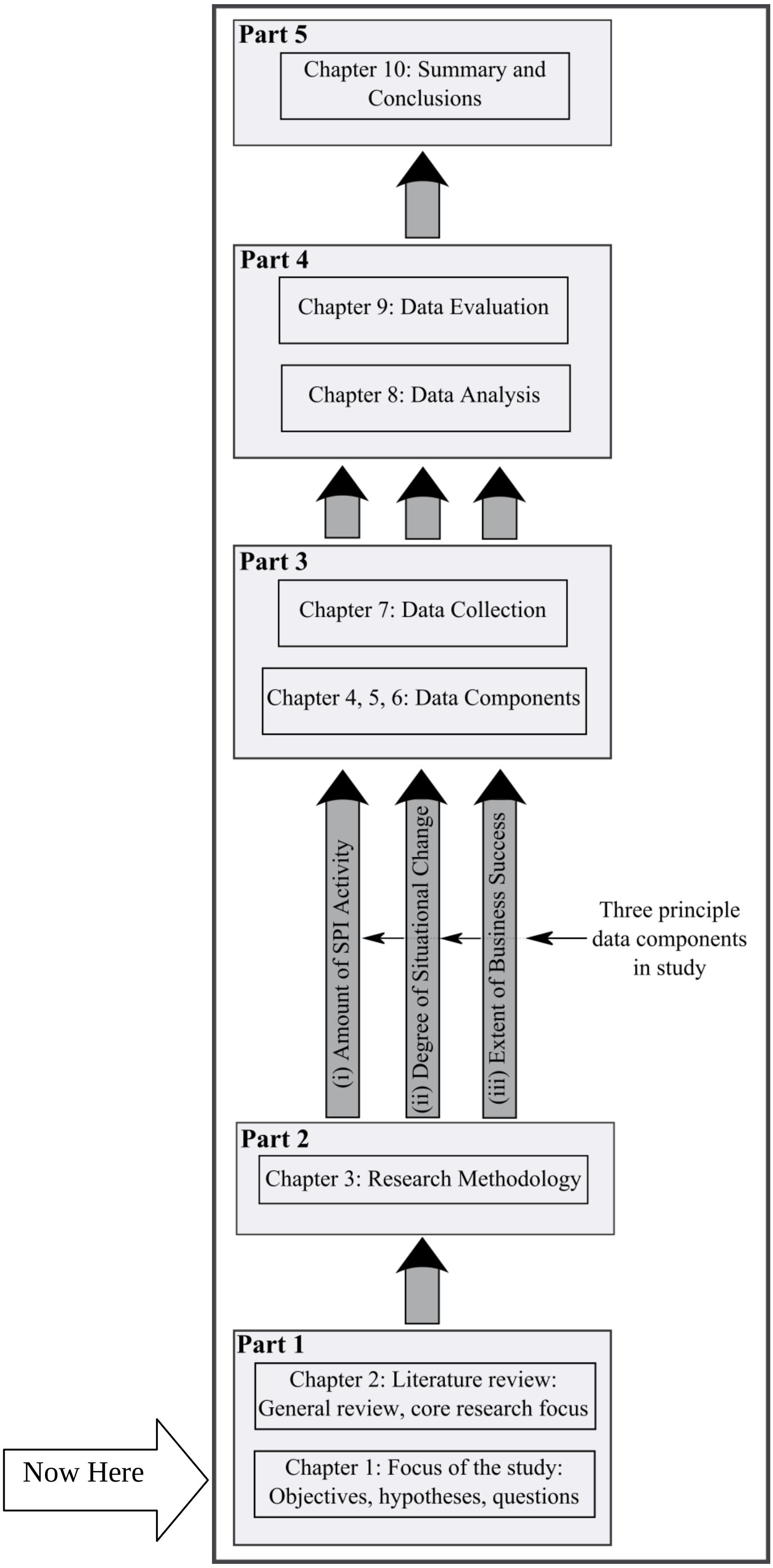


Figure 1 Map of Thesis – Part 1

Part 1 Study Background

The first part of the thesis contains two chapters. Chapter 1 outlines the principle motivation for conducting this research, along with a description of the research objective. The research hypotheses and questions are also identified in chapter 1. In chapter 2, the general field of software processes and SPI is introduced. The latter sections in chapter 2 are concerned with the related literature with respect to SPI in software SMEs, and with identifying the core research focus.

1 Focus of the Study

1.1 Introduction

This chapter provides a high level background to the research, followed by a brief description of the research objective. Thereafter, research hypotheses and questions are the presented, followed by a breakdown of the structure of this thesis.

1.2 Motivation

Business processes are the logical organisation of people, materials, energy, equipment, and procedures into work activities designed to produce a specified end result (work product) (Pall 1987). Such processes are considered to be important because they affect the quality of the products and services that the organisation delivers to its clients; with an effective business process producing high-quality products in a cost effective way (Cugola 1998). While some debate exists in relation to the degree of importance of business processes in enabling business success (Vergidis, Tiwari and Majeed 2008), various empirical studies have supported the view that business processes (and business process improvement) are important for business performance (McCormack and Johnson 2001, Skrinjar, Bosilj-Vuksic and Indihar-Stemberger 2008). Moreover, the importance of business processes is acknowledged by the inclusion of business process performance perspectives in many of the contemporary business performance measurement frameworks (Kaplan and Norton 1992, Neely, Adams and Kennerley 2002).

In software development organisations, the software development process is a large and complex component of the overall business process and therefore, along with other business processes, effective management of the software development process should support the achievement of business success. SPI involves the improvement of the software development process and is therefore concerned with the effectiveness of the process. However, to date the relationship between SPI and business success in software SMEs has received little attention, with no earlier published study being dedicated directly to an in-depth examination of the relationship between SPI and business success in software SMEs. As a result, there is a lack of solid empirical data regarding the relationship between the amount of SPI activity and business success in software SMEs. Business processes, including the software development process, exist so as to support a business in fulfilling its objectives. Business process improvement, including SPI, is concerned with improving the method of work, which should ultimately have the consequence of improved business success. Therefore, SPI should be positively associated with business success. However, for software SMEs, there is no empirical evidence supporting this relationship at present. Consequently, this researcher believes that it is important to examine if SPI activity is having a positive effect on business success in software SMEs.

1.3 Research objective

SPI is concerned with improving the software development process, with the goal of supporting the software development effort in producing and maintaining software products, hence benefiting business survival and success prospects. The empirical study associated with this thesis has been designed to systematically examine the influence of SPI in software SMEs. *The objective of the study is to investigate if SPI is positively associated with business success in software SMEs.*

Specifically, this research is concerned with profiling the amount of SPI in organisations and examining if there is a statistically significant association between SPI and business success across multiple software SMEs. Furthermore, in examining SPI, it is the intention of this study to cover the broadest possible range of software processes, and to consider the full extent of possible changes to each process. Such process changes can be both large, formal SPI initiatives and small, informal modifications to the work process. Since each organisation operates in a different environment and different organisations experience varying degrees of situational change at different stages of business development, the research is also concerned with investigating the amount of situational change that is occurring in each of the participating companies.

This researcher believes strongly that the degree of success achieved by software SMEs is influenced by the quality of the software development process and the ability of software development organisations to adapt their development processes to appropriately address emerging needs. A review of the SPI literature indicates that there is insufficient evidence to support this belief at present, especially with regard to the importance of actively evolving software development processes to address changing situations.

Since earlier related SPI research has reported that software SMEs can have a low process priority (Baddoo and Hall 2003), tending to implement SPI in response to negative business events (Coleman and O'Connor 2008), this researcher considers it vitally important that an empirical investigation explores the relationship between SPI and business success. Although theory and to some extent common sense inform us that effective SPI should be positively associated with business success, it is perhaps the lack of empirical evidence of such a relationship that results in a low process priority in software SMEs. Therefore, the collection, analysis and evaluation of SPI and business success data from a number of software SMEs will enable the study to make strong, empirically-based statements concerning the relationship between SPI and business success. Furthermore, by also empirically examining the extent of situational change, an additional degree of robustness can be claimed in the research findings.

1.4 Research hypotheses and questions

Having completed an extensive review of the related published research (which is summarised in chapter 2), and with a view to the central objective of this study, two research hypotheses were developed. These hypotheses provide a clear focus for the investigation.

> ***H1:*** *SPI activity is positively associated with business success in software SMEs.*

> ***H2:*** *To maximise business success, SPI activity should be in proportion to the degree of situational change.*

There are a number of possible outcomes when testing the hypotheses in the study. Firstly, if either hypothesis is proven true, then evidence of a positive relationship between SPI and business success would be established. As outlined in the earlier part of this chapter, evidence of the benefits of SPI is presently lacking and therefore, to find that either hypothesis is true would represent a significant addition to the argument in favour of SPI in software SMEs. Secondly, the evidence gathered in this study could disprove either hypothesis. In the case of H1, this second outcome would represent evidence that SPI does not tend to have a positive influence on business success – or even that increases in SPI have a negative association with business success. To disprove H2 would suggest that the degree of situational change is not an important factor when implementing SPI in software SMEs. Irrespective of the outcome of the hypotheses testing, the study will advance our understanding of SPI in practice in software SMEs, the relationship between SPI and business success in software SMEs, and the influence of situational change in software SMEs.

The two research hypotheses under investigation in this study are concerned with three distinct phenomena in software SMEs: the amount of SPI activity, the extent of business success, and the degree of situational change. Through examining these phenomena in detail, it is also possible to develop a deeper understanding of software development in SMEs. In particular, the research will facilitate the development of a greater understanding of the meaning of business success for software SMEs, along with the preferred SPI actions in SMEs. Furthermore, it will also be possible to develop an understanding of the changing nature of situational contexts in software SMEs. In order to focus the investigation of these three phenomena, three research questions have been developed for the study:

> ***RQ1:*** *What is business success for software SMEs?*
>
> ***RQ2:*** *What aspects of the software development process are commonly undergoing SPI in software SMEs?*
>
> ***RQ3:*** *What aspects of the situational context are commonly experiencing change in software SMEs?*

Since no earlier published study has comprehensively examined either business success or situational change in software SMEs, the answers to RQ1 and RQ3 will establish for the first time the extent and type of business success and situational change that is occurring in software SMEs. In the case of RQ2, it will be possible to compare and contrast the research findings with the reports from earlier studies concerning the more common or most important process improvement activities for software SMEs. This will facilitate the confirmation of findings from previous studies as well as offering the possibility of discovering new areas that are important for SPI in practice in software SMEs.

1.5 Research process

The general research process adopted in the present study has been adapted from the approach proposed by Oates (2006). As depicted in Figure 2, the research process consists of five sets of activities that provide a structure for the research programme:

1. *Initiation* – concept creation and elaboration involving a review of the related literature and the development of research hypotheses and questions.
2. *Research paradigm* – determine a suitable research paradigm within which the research can be actualised.
3. *Data generation* – identification of a suitable, robust and reliable approach, or suite of approaches, to satisfying the research data requirements.
4. *Data analysis and evaluation* – identification of appropriate and effective research data analysis and evaluation methods.
5. *Examination of research hypotheses and questions* – examine the research hypotheses based on the data analysis and evaluation; provide answers to the research questions.

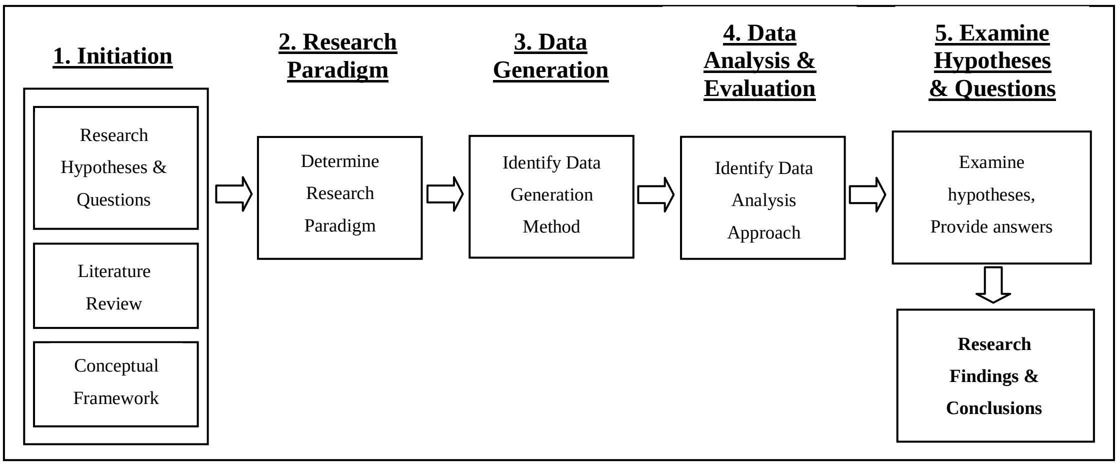


Figure 2 Model of research process (adapted from Oates, 2006)

Throughout this research, the overall research process identified in Figure 2 was adopted. Consequently, the five major parts of this thesis (as presented in Figure 1) are broadly reflective of the five steps outlined in Figure 2. The following section presents a more detailed breakdown of the structure of this thesis, including the contents of the individual chapters.

1.6 Structure of this thesis

The remaining chapters of this thesis are structured as follows. Chapter 2 presents the related literature in the software process and SPI domains. The research methodology and study design are outlined in chapter 3, including details of the data generation mechanisms.

Chapter 4 outlines the approach to examining the amount of SPI activity, with chapter 5 providing details of the investigation of situational change. In chapter 6, the approach to examining the extent of business success is presented. Chapter 7 provides details of the study timeframe, along with details of the participating organisations and the actual data collected.

In chapter 8, the data analysis is presented. This includes a detailed account of the methods for calculating the amount of SPI, the degree of situational change, and the extent of business success. Chapter 8 also outlines the correlations conducted on the data, while chapter 9 performs a more in-depth evaluation of the data analysis results.

Chapter 10 provides a summary and conclusion, wherein the contribution of the research is examined, along with an evaluation of the impact of the research findings on the broader field of SPI. In addition, chapter 10 provides a description of the acknowledged limitations of the research, with a closing section outlining possible areas for future research.

1.7 Description of terms

This thesis introduces three new terms, which are used extensively throughout the research. Therefore, this section explicitly defines these terms, along with a brief description of their meaning.

SPI Activity is considered by this thesis to be the amount of SPI implemented in an organisation over a period of time. Furthermore, in measuring SPI activity, this study considers a broad spectrum of possible SPI implementations, ranging from large formal SPI initiatives to small informal software process improvements. For the purpose of this study, SPI activity is examined over a 12 month period, using a survey instrument which retrospectively assesses changes to the various different dimensions of the software development process. Further details are presented in chapter 4.

Situational Change is considered by this thesis to be the extent of change in the factors that affect the software development process over a period of time. For example, if the requirements are becoming more volatile (i.e. more subject to change) then the software development process may need to be adapted in order to be more capable of responding to changing or emerging requirements. In measuring situational change, a broad spectrum of different types of change are considered, ranging from a significant change to a situational factor that affects the software development process to a minor change to a situational factor. For the purpose of this study, situational change is examined over a 12 month period, using a survey instrument which retrospectively assesses changes to the various situational factors that can affect the software development process. Further details are presented in chapter 5.

Business Success is considered by this thesis to be the extent of achievement of business objectives over a period of time. For the purpose of this study, business success is examined over a 12 month period using a two-phased engagement. In the first engagement, at the start of the 12 month period, the business objectives are elicited and formally recorded for the forthcoming 12 months. At the end of the 12 month period, the extent to which the recorded objectives have been achieved is assessed. Following this latter assessment, overall organisational business success measures are determined. Further details are presented in chapter 6.

2 Literature Review

2.1 Introduction

The first section of this chapter outlines the general approach adopted whilst conducting my literature review. Thereafter, the related software process and SPI literature is presented. Since this research is focused on software SMEs, this chapter then undertakes a review of the literature regarding software processes and SPI in software SMEs. The final part of this chapter is dedicated to describing the core research focus.

2.2 Approach to literature review

While the study was principally motivated to examine the relationship between SPI activity and business success in software SMEs, it was the determined resolution of the researcher to first develop a broad understanding of the general literature in relation to software development processes and SPI. In order to pursue this resolution, from the outset, a highly structured approach was adopted in relation to performing the literature review. The first step in the literature review was to analyse the central areas related to the research. The initial domains for investigation included the software process in general, SPI, process models and methodologies, process maturity frameworks, process quality standards, and classical and contemporary approaches to software development. This initial pass at the related research consisted of almost 500 papers and formed the basis of my literature archive. Naturally, a few key areas emerged as being of particular interest, including software process maturity, empirical SPI studies, and software development in software SMEs.

To assist the initial pass of the literature, the established repository of SPI-related publications assembled over many years by my supervisor – an acknowledged expert in the proposed area of research – was studied in detail. From this exercise, the closely related research themes became evident, with many of the key publication fora and authors emerging during this time. The traditional literature reviewing skills of further investigating emerging authors, publications and references permitted a more in-depth analysis of the key areas of interest to this particular study. Gradually, a clearer understanding of all the core related research domains and contributions was developed through the continual application of these traditional techniques for literature reviewing, with the landscape of related domains becoming more evident as the literature review was expanded. The closely related domains that became apparent at this time included: SPI (in general), the role of standards for SPI, the success and failure factors associated with SPI initiatives, software processes and SPI in small organisations, and project management and success.

Whilst growing the base literature archive, it became evident that a number of major conferences were consistently publishing research that was of direct relevance to my research area, including: European Systems & Software Process Improvement (EuroSPI), Software Process Improvement and Capability dEtermination (SPICE), Product-Focused Software Development and Process Improvement (PROFES), the International Conference on Software Engineering (ICSE), and the Euromicro conference on Software Engineering and Advanced Applications (SEAA). During the same period, a number of journals also presented as being key sources for closely related research, including the Journal of Systems and Software (JSS), the Journal of Software Maintenance and Evolution: Research and Practice (including the former Journal of Software Process: Improvement and Practice), the Information and Software Technology journal (IST), Crosstalk – the Journal of Defense Software Engineering, IEEE Transactions on Software Engineering (TOSE), IEEE Software, IEEE Computer, and Communications of the ACM. For all of these publications, and some others that were also considered to offer sources of related research, issue alerts were established so that once the initial literature review was discharged the practice of staying up to date with the literature would be simplified. Both the initial literature review and the study-long effort to identify emerging related works involved the analysis and review of research papers, and where appropriate following-up references that were considered to be of relevance.

Throughout the literature review phase, papers that were identified as being of interest were classified, noted in a working notes document, and entered into a hosted reference manager application.[1] At the end of the initial literature search, the database contained in excess of 700 categorised papers. This database is intended as a longer

[1] The RefWorks ® reference manager product was employed.

term repository for research areas that are of interest to the researcher, and therefore not all of the papers are directly relevant to this PhD thesis. However, of the 700+ papers in the reference database, over 350 papers were deemed to be relevant to this PhD thesis, and for each of these papers, a brief summary was maintained for future reference.

Having completed the initial literature review and classification phase, the publications and authors of central interest were reviewed in detail in order to ensure a complete understanding, with further additional following-up of the references contained within these papers. Additional online searches were performed using standard search engines, to ensure that all material of central relevance to the research theme was reviewed. In particular, search strings such as "*Software process improvement/SPI business success*", "*Software process improvement/SPI situational/context change*", "*Software process improvement/SPI benefit(s)/advantage(s)*", "*Software process improvement small company/organisation/SME"*, "*Software process improvement", "Business success software company/organisation/SME".*

Throughout the initial literature review cycle, mind maps were developed and extended to support the general understanding of the relationships between the various related research domains. The high level mind map is presented in Figure 3, with the lower level mind map for the SPI domain presented in Figure 4.[2] Further mind maps, which demonstrate the expansive nature of the literature review, are presented in Appendix A.

A literature review document was drafted following the completion of the initial review. So as to ensure that the review was kept up to date, any related publications that were identified throughout the research were added to the literature review document. In the following sections, the resultant literature review is presented, starting with the general domain of software process and process improvement.

---

[2] The right-facing arrows in Figure 3 and Figure 4 indicate that the mind map is further extended at this point (with some of the extended mind maps presented in Appendix A).

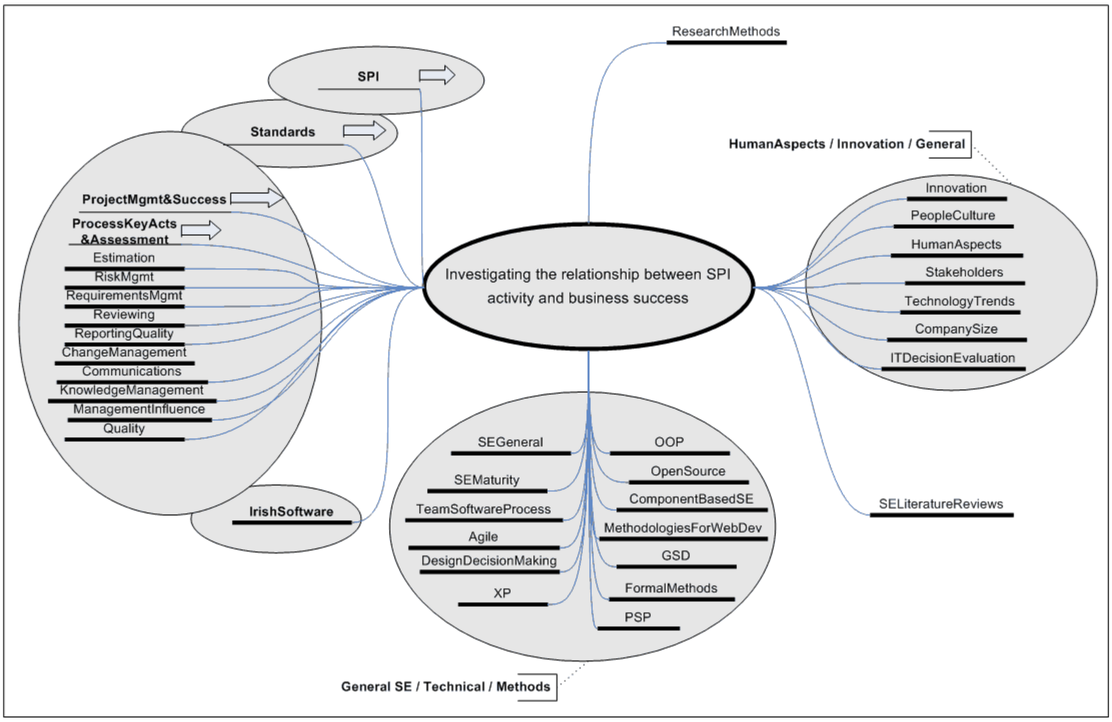


Figure 3 Overview of literature review – mind map

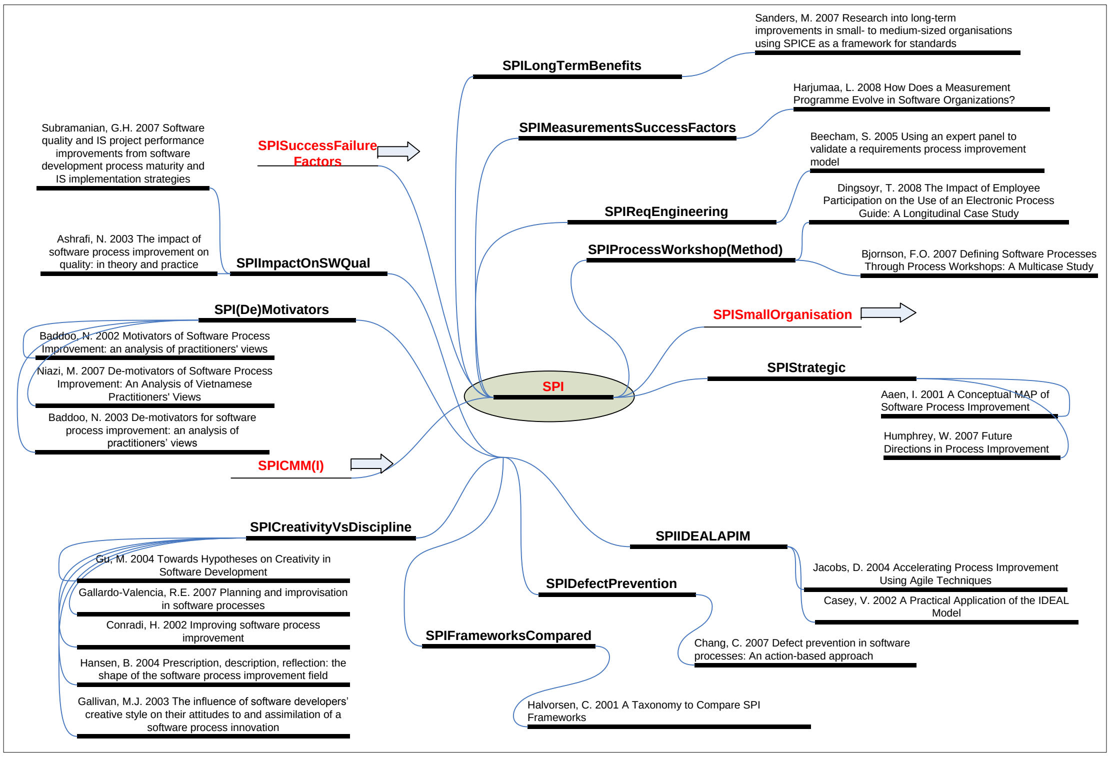


Figure 4 Overview of related SPI literature – mind map

2.3 Software process and process improvement

This section examines the domain of software processes and SPI. A number of software lifecycle models are presented, followed by a review of some established SPI frameworks. Thereafter, the field of agile software development is examined. This section concludes by outlining the influence of situational context on software processes and SPI decisions.

2.3.1 Software process and SPI

According to the noted software process authority Watts Humphrey, the software process is defined as "*the sequence of steps required to develop or maintain software*" (Humphrey 1995 p4). Within this sequence of steps, there exist multiple activities, each of which, depending on the needs of the individual software development setting, can be implemented to varying degrees. A comprehensive listing of all potential software process actions is provided by the International Organisation for Standardisation (ISO) and the International Electrotechnical Commission (IEC) in ISO/IEC 12207 (ISO/IEC 2008), which identifies 43 individual processes that are classified under 7 different categories:

- *Agreement Processes* – relate to the activities required to establish agreement between two organisations engaged in the exchange of products or services.
- *Organisational Project-Enabling Processes* – relate to the organisation project management activities required to conduct a software project.
- *Project Processes* – relate to the planning, assessment and control of projects.
- *Technical Processes* – relate to the activities required to translate user needs into working deliverables in a repeatable fashion and the eventual retirement of products.
- *Software Implementation Processes* – relate to the production of system components.
- *Software Support Processes* – relate to a group of specialised processes that support the software implementation processes.
- *Software Reuse Processes* – relate to the development and life time management of reusable components.

Owing to the differences that exist between individual software development settings, there is no one ideal software development process that is generally applicable to all software development efforts (Sommerville 2007, Jones 2008). Within individual software development settings, critical items such as the development technologies, personnel, requirements and clients are continually undergoing change. As a result, an effective development process is one that is not set in stone but rather one that dynamically responds to environmental feedback (Zahran 1998). Consequently, software development teams and managers may reflect on the development process and undertake SPI so as to "*create more effective and efficient performance of software development and maintenance through structuring and optimising of processes*" (Van Solingen 2001 p455).

The present SPI landscape has evolved over many decades, drawing inspiration from a variety of sources. The concept of controlling the quality of products has long existed in the manufacturing industry. In the 1930s, Walter Shewhart introduced the principle of statistical quality control of manufactured products (Shewhart 1931). Shewhart's work was further developed by William Edwards Deming and Joseph Duran in the 1950s, with Deming's contribution, including the *Plan, Do, Check, Act* approach to continuous improvement (Arthur 1993), enabling Japanese industry to excel at the delivery of quality products (Aguayo 1991). In the 1970s, Philip Crosby introduced the concept of preventative quality control in product development environments and presented an organisational maturity matrix (Crosby 1979).

As software development projects have increased in number and in complexity, some of the principles of industrial quality management have been introduced into the software development process. A number of the contemporary software development approaches, such as the Capability Maturity Model Integrated (CMMI) (SEI 2006) and ISO/IEC 15504 (ISO/IEC 2005), incorporate a process maturity dimension. Other software development approaches, such as ISO 9001 (ISO 1994), bear the influences of manufacturing-based process control. More recently, in response to the challenge of rapidly changing software development requirements, agile approaches have been proposed as alternative SPI vehicles (Aaen, Borjesson and Mathiassen 2007). Such agile approaches include XP (Beck 1999) and Scrum (Schwaber 1995, Schwaber and Beedle 2002).

There exists a broad variety of SPI approaches, far beyond the few that have been listed in the previous paragraphs. And although some debate exists regarding the extent of the benefits bestowed by individual SPI approaches, it is generally accepted that the quality of the software development process has a direct influence on the quality of software products (Humphrey 1995, Sommerville 2007, Zahran 1998).

While software development processes are inherently complex and can involve very large numbers of activities (Sommerville 2007), as is evident in the 43 processes in ISO/IEC 12207, there are four fundamental activities that are generally required (Sommerville 2007):

- *Software specification* – identifies the functionality.
- *Software design and implementation* – the production of the software as per the specification.
- *Software validation* – the software is assessed to ensure that it complies with the specification.
- *Software evolution* – the adaption of the software to satisfy changing customer needs.

Simplified descriptions of the software development process called software development lifecycle models provide a useful framework from which to build more detailed process descriptions.

2.3.2 Software development lifecycle models

A software development lifecycle model is an abstract representation of the software development process as "*a sequence of stages that may overlap and/or iterate, as appropriate for the project's scope, magnitude, complexity, changing needs*" (ISO/IEC 2008 p12). Lifecycle models outline a general philosophy with respect to the software development process and in practice, multiple lifecycle models can be used in conjunction to deliver a software product (Sommerville 2007). The following generic models are in common use (Sommerville 2007):

- The Waterfall model.
- Evolutionary development.

- Component-based software engineering

The Waterfall model (Royce 1970) consists of five stages (requirements capture, system and software design, implementation, testing and evolution) that must be completed in strict sequence. Evolutionary development, for which the Spiral model (Boehm 1988) is an example, involves using the basic Waterfall steps to gradually produce and evolve an implementation of a complete system through a process of user feedback. Component-based software engineering (Heineman and Councill 2001) involves lower risk and faster implementation of software systems by re-using previously developed components that are suited to the needs of the current implementation. Lifecycle models, such as those identified above, provide a framework from which to build a more detailed software development process. However, the specifics of a software development process are much more complex and the process effectiveness is dependent on not just the suitability of the process definition, but also on the discipline with which it is discharged.

2.3.3 Process discipline, maturity and capability

All software development efforts follow a process (Van Vliet 2000), the details of which may be defined to varying degrees or not explicitly described at all. Where the process is not documented, participants in the development effort depend on verbal communication in order to understand the method of work. Although effort is required to document a process, there are many benefits to having an explicit definition, including: assisting the process implementation, making the process effective and producing the desired process outcomes (Dyba, Dingsoyr and Moe 2004).

An effective process supports common process thinking, which facilitates disciplined behaviour across team members (Zahran 1998). Where process discipline exists, individual goals can be more easily aligned with team goals. And the successful fulfilment of team goals enhances the chances of successful project completion. A team member has *internalised* the process when they apply the process definition as an automatic action; an organisation has *institutionalised* the process when all members follow the process definition and process discipline is enforced (Zahran 1998).

Software process maturity is "*the extent to which a specific software process is explicitly defined, managed, measured, controlled and effective*" (Paulk et al. 1993 p4)[35]. Higher levels of process maturity are associated with higher product quality, reduced production costs (Harter and Slaughter 2003), and with increased predictability of the process results (Ferguson et al. 1999, Harrison, Settle and Raffo 2001). Therefore, more mature organisations are said to have a higher process capability. One method of improving the maturity of the software development process is to employ the use of a process maturity reference framework.

2.3.4 Process maturity reference frameworks

Any SPI strategy should be based on an overall framework (Zahran 1998). Process maturity reference frameworks provide a structured and proven process improvement path by encapsulating the accumulated experience from many projects and domain experts. Consequently, such frameworks incorporate established best practice. The purpose of this section is to present an overview of the structure and benefits of two of the most widely recognised process maturity reference frameworks: CMMI (SEI 2006) and ISO/IEC 15504 (ISO/IEC 2005).

2.3.4.1 CMMI

In 1984, the U.S. Department of Defense (DoD) established the Software Engineering Institute (SEI) as a government-funded software engineering research and development centre (SEI 2009b). Owing to the difficulties experienced in the delivery of many large software projects, the SEI created the Capability Maturity Model (CMM) (Paulk et al. 1993) to assist the U.S. federal government in assessing the capabilities of its suppliers in the software development area (Boehm and Turner 2003, Adler 2005). The CMM evolved over time, eventually being replaced by an extended version, the CMMI (SEI 2006).

The CMMI identifies a process roadmap which software development organisations can adopt in order to achieve higher levels of process maturity. It consists of a comprehensive list of the Key Process Areas (KPAs) that are required for the production and lifetime support of software. The CMMI is applied in one of two mutually-exclusive modes – *staged* or *continuous*. The *staged* framework offers a tried and proven path for success, consisting of five cumulative plateaus of organisational maturity (*Initial, Managed, Defined, Quantifiably Defined* and *Optimised*), which characterise a broad spectrum of software process implementations ranging from the chaotic to the continually improved. Maturity ratings are awarded in the *staged* representation and in order to secure some software contracts, a minimum CMMI rating may be required (Van Vliet 2000). Rather than rating the organisational maturity as a whole, the CMMI *continuous* framework provides an individual rating for each process area according to a six point scale (*Incomplete, Performed, Managed, Defined, Quantifiably Managed* and *Optimised*). The *continuous* framework doesn't provide a maturity rating, however, it does provide a process profile that can be used in contract tendering.

Numerous studies have highlighted the benefits of CMMI adoption, especially in large organisations. The benefits include productivity gains and reductions in post-release defects (Herbsleb et al. 1994, Herbsleb and Goldenson 1996), successful project completion (Lawlis, Flowe and Thordahl 1995), improvements in the quality of working life and organisational learning and efficiency (Hyde and Wilson 2004), and reductions in the overall cost of quality (Gibson, Goldenson and Kost 2006). However, there may be a non-uniform delivery of benefits at different levels of process maturity, with some evidence suggesting that there is a diminished cost-benefit proposition beyond level 3 (Jiang et al. 2004). This diminished value proposition may be one reason why just 12.7% of companies have progressed beyond the Level 3 (SEI 2009a).

Some studies have demonstrated that CMMI can be beneficial for smaller organisations and projects (Cepeda, Garcia and Langhout 2008, Serrano, Montes de Oca and Cedillo 2005), especially if process selection and adaptation are tailored to the needs of the particular setting (Paulk 1998, Leung and Yuen 2001, Jost 2008). However, since smaller companies can be challenged with customer pressure and a general lack of time and resources, it has also been claimed that the CMMI may not ideally suited to smaller software development companies (Leung 1999, Staples et al.

2007, Miluk 2005, Saastamoinen and Tukiainen 2004, Khurshid, Bannerman and Staples 2009), an observation which has motivated other studies to attempt to identify the high-value CMMI practices for smaller software development settings (Wilkie, McFall and McCaffery 2005, Niazi, Babar and Ibrahim 2008, Niazi and Babar 2009).

The studies identified above highlight that the CMMI can be beneficial for larger companies, with some disagreement on the extent of the suitability of CMMI for smaller software development companies. However, while benefits have been demonstrated in both large and small settings (including improvements in total cost of development and quality of outputs), none of the earlier published studies have attempted to examine the relationship between CMMI and the broad spectrum of business success considerations for software development companies.

2.3.4.2 ISO/IEC 15504

ISO/IEC 15504 (ISO/IEC 2005) is a series of nine individual standards that are maintained by Joint Technical Committee 7 of the ISO and IEC. In ISO/IEC 15504, process capability is expressed using a two-dimensional model that is similar in layout and composition to the continuous representation of CMMI (which enables an easy comparison between the two process maturity reference models). Owing to its origins in the Software Process Improvement Capability dEtermination project (Dorling 1993), ISO/IEC 15504 is often referred to as SPICE.

The ISO/IEC 15504 process listing is based on ISO/IEC 12207 (ISO/IEC 2008), with ISO/IEC 15504 classifying the process components under five categories: *Customer-Supplier*, *Engineering*, *Supporting*, *Management* and *Organisation*. A capability rating, which reflects the extent to which the process achieves its defined purpose and objectives (Sanders 1998 p57), is provided for each process according to a six-point scale (*Incomplete*, *Performed*, *Managed*, *Established*, *Predictable* and *Optimising*). That ISO/IEC 15504 *"is controlled and regularly reviewed in the light of experience of use, and is changed only by international consensus*" (Barafort, Di Renzo and Merlan 2002 p319) is generally considered in a positive light.

Early evidence of the benefits of ISO/IEC 15504 was gathered in the SPICE trials (El Emam 1998, Jung et al. 2001, Galin 2004), with related studies demonstrating that ISO/IEC 15504 was beneficial for large organisations (Jarvinen, Hamann and Van Solingen 1999), especially for improving customer satisfaction and project performance (Wegelius and Johansson 2007, El Emam and Birk 2000). However, since situational circumstances are continually changing, it has been suggested that ISO/IEC 15504 would benefit from the addition of a continual process feedback and analysis dimension (Jarvinen, Hamann and Van Solingen 1999, Lepasaar, Varkoi and Jaakkola 2001).

While some studies suggest that ISO/IEC 15504 can be beneficial for small software development companies (Cater-Steel 2001, Anacleto et al. 2004), other research findings raise some issues concerning the suitability of ISO/IEC 15504 for small software development settings (Jung et al. 2001, Laporte et al. 2005). The concerns centre on the complexity of the standard and the cost of implementation, and essentially, these are criticisms that can be levelled at process maturity in general. Consequently, it has been observed that "*measures of capability are not suitable for small organizations, or that software development process capability has less effect on project performance for small organizations*" (El Emam and Birk 2000 p119). And it may not just be small organisations that are unsuited to process maturity frameworks and concepts – with some research claiming that fewer than 5% of all software development enterprises use any type of software maturity model (Magee and Thiele 2004). However, while there exist concerns in relation to the applicability of process maturity models to small companies, of the two best known process maturity reference frameworks, the available literature suggests that ISO/IEC 15504 can be scaled for use by small companies more easily than CMMI (O'Connor and Coleman 2007).

Although there is considerable divergence of opinion regarding process maturity frameworks, the related research suggests that in general they can work for some (though not all) large companies and that they are perhaps unsuited for implementation in most (though not all) small software development settings. The research to date for both CMMI and ISO/IEC 15504 has focused on examining the benefits of implementation, including improvements in predictability and quality of project deliverables. However, the earlier research has not focused on examining the effect that process maturity frameworks have on general business success. Perhaps if evidence of a positive association with business success were to be established, more organisations might consider process maturity framework implementation.

This section has focused on examining process maturity frameworks and establishing what the research to date can tell us about the successes and shortcomings of two of the leading process maturity frameworks: CMMI and ISO/IEC 15504. Process maturity reference frameworks such as ISO/IEC 15504 and CMMI are concerned with the concept that processes can be improved, offering higher levels of capability. However, other approaches exist for the production of software that are less concerned with the process capability dimension. The ISO 9000 quality standard (ISO 2000a) is one such approach.

2.3.5 ISO 9000 quality standard

Published in the 1980s and revised over time, ISO 9000 is a family of generic quality management standards that can be applied to any type of organisation in order to address customer and regulatory requirements. This is achieved through the adoption of a systematic approach to managing the organisation's processes so that they consistently turn out products that satisfy customer expectations (ISO 2009a). Within the ISO 9000 series, it is the ISO 9001 standard (ISO 2000b) that ensures that suppliers conform to specified requirements throughout the development cycle, with ISO 9000-3 (ISO 2004) aiding the application of ISO 9001 to software development environments. Many companies across different sectors have adopted ISO 9000 and over 950,000 ISO 9001 certificates have been issued in 175 countries (ISO 2008).

ISO 9000 standards involve addressing all aspects of development that may affect the quality of the final product, ensuring that the development process is clearly documented and implemented (Schuler 1995). ISO 9001 certification involves the participation of an external and independent auditor, whose responsibility it is to confirm that the process is explicitly documented, along with credible evidence of process adherence. Hence, ISO 9001 certification indicates that an organisation can consistently transfer inputs to outputs with a high degree of quality (Zahran 1998).

However, although obtaining certification can help an organisation to "*avoid multiple audits by its clients*" (ISO 2009a), the cost of obtaining certification is the subject of persistent criticism (Stevenson and Barnes 2001).

One of the key strengths of ISO 9001 is that it ensures that an organisation documents and follows a software development process (Coallier 1994) – though the very act of documenting the process is acknowledged as being quite onerous (Schuler 1995). Furthermore, the benefits of adopting ISO 9001 are considered to be greatest where the pre-existing process definition was weak (Stelzer, Mellis and Herzwurm 1997). Although earlier research suggests that the significant majority of organisations that adopted ISO 9000 believed that the benefits exceeded the implementation cost (Stelzer, Mellis and Herzwurm 1996), there are two principle and recurring criticisms of ISO 9000: it is deficient regarding the specifics of software development (Stelzer, Mellis and Herzwurm 1996, Yang 2001, Oskarsson and Glass 1996) and it lacks a continuous improvement dimension (Coallier 1994, Grady 1997). Since software development is dynamic, evolutionary and innovative in nature, a continuous improvement focus is considered by many to be a critical ingredient of a software development process.

The ISO acknowledges that the benefits of ISO 9000 are more obvious for larger businesses, where efficient resource utilisation requires that "*written procedures, instructions, forms or records help [to] ensure that everyone is not just doing his or her own thing*" (ISO 2009b). While these benefits may be attractive to large organisations, it has been suggested that ISO 9000 is overly bureaucratic for smaller software development companies (Stelzer, Mellis and Herzwurm 1997), that it represents a disincentive for creativity (Blind and Hipp 2003) and "*a source of organisational rigidity that can hamper innovation*" (Grimaldi and Torrisi 2001 p1438). Indeed, the difficulty of applying ISO 9000 in small organisations is noted even outside of the software development business (Graham 1994, Clifford 2005).

This section has examined the suitability of ISO 9000 for software development companies, finding that there can be benefits to implementing the standard in larger organisations. However, along with ISO/IEC 15504, ISO 9000 is sometimes criticised for lacking a continuous improvement mechanism. This criticism is significant as software development companies continually need to adapt the software development process so as to address changes in the situational context. The published ISO 9000 and ISO/IEC 15504 related research is also limited insofar as there is an absence of studies that examine the role of such standards and frameworks in supporting business success.

2.3.6 Process assessment and process audit

As discussed in the preceding sections, software process maturity frameworks and quality management standards represent two distinct approaches to software development. As a result, they also adopt two different mechanisms with respect to evaluating the software development process: CMMI and ISO/IEC 15504 use process assessments while ISO 9001 uses a process audit. Process assessments compare *"the actual performance of a process to some form of structured process model that serves as a yardstick*" (Rout et al. 2007 p1484) and result in the award of a process maturity rating. Process audits are concerned only with conformance (Rout et al. 2007) and take the form of "*an independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements and other criteria*" (IEEE 1991 p23). While audits are largely focused on certification, assessments have an extended function in that they involve "*a review of a software organisation to advise its management… on how they can improve their operation*" (Humphrey 1989 p36). Assessments are therefore naturally aligned with process maturity reference frameworks, with CMMI and ISO/IEC 15504 each having their own particular take on the assessment configuration.

Within the CMMI framework, the term *appraisal* is synonymous with the term *assessment* and is defined as "*an examination of one or more processes by a trained team of professionals using an appraisal reference model as the basis for determining, at a minimum, strengths and weaknesses*" (SCAMPI Upgrade Team 2006 p27). The Appraisal Requirements for CMMI (ARC) "*defines the requirements considered essential to appraisal methods intended for use with CMMI models*" (SCAMPI Upgrade Team 2006 pvii). Under the provisions of ARC, three different appraisal classes are established (Class A, B and C). The key differentiating attributes for appraisal classes are "*the degree of confidence in the appraisal outcomes, the* *generating of ratings and the appraisal cost and duration*" (SCAMPI Upgrade Team 2006 p5). In the case of ISO/IEC 15504, the combined use of ISO/IEC 15504-2 (ISO/IEC 2003) and ISO/IEC 15504-3 (ISO/IEC 2004) permits the discharge of process assessments that produce process profiles consisting of a set of process ratings and, optionally, a capability level for each process assessed.

ISO 9000 identifies the requirements that quality systems must meet, but it does not indicate how these requirements should be met in any particular organisation (ISO 2009a). As a result, each organisation that implements ISO 9000 establishes a documented quality management process, with an audit being required in order to determine the suitability of the process and to establish the degree of conformance. External audits, those carried out by an independent third party, are required in order to obtain an ISO 9000 certificate of conformity. Certification is valid for a three year period, after which a company must successfully complete a further external audit carried out by an accredited quality system certification body (Terlaak and King 2006).

Although certification is one of the major benefits of ISO 9000, it has been observed that obtaining certification is costly (Van Der Wiele and Brown 1997), a reality that may act as an impediment to ISO 9000 implementation in smaller and more resource-bound companies. Equally, CMMI and ISO/IEC 15504 assessments may not suited to small software companies that require process assessment techniques that are agile, quick and inexpensive (McCaffery, McFall and Wilkie 2005). As a result, a set of process assessment methods have been designed specifically to help small organisations (Cignoni 1999, McCaffery, Taylor and Coleman 2007, Cater-Steel 2002, Habra et al. 2002, Pino et al. 2010). Lightweight process assessments attempt to complete effective, accurate and useful assessments, with evidence of the utility of such approaches established in a number of studies (Habra et al. 2002, Peterson et al. 2002). It has, however, also been suggested that additional research into the efficacy of light weight assessment approaches is required (Mishra and Mishra 2009).

Software development reference models and quality standards, and the associated assessment vehicles, represent significant contributions to the field of software development. However, for reasons that are likely to be related to bureaucracy, inflexibility and cost, it appears that such approaches are not widely implemented in software SMEs in practice. As a community, we therefore need to examine how the needs of smaller software development organisations can be addressed either through the extension of existing approaches or through the creation of new mechanisms for improving the performance of software development processes. Over the past decade, one such new mechanism that has emerged falls under the general title of *Agile* software development.

2.3.7 Agile software development

Sometimes referred to as agile methodologies, agile software development methods are a family of approaches to software development that are based on a set of principles encapsulated in the four directives of the Agile Manifesto (Fowler and Highsmith 2001): *Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation; Responding to change over following a plan*. Some of the more common methods include eXtreme Programming (XP) (Beck 1999), Scrum (Schwaber 1995, Schwaber and Beedle 2002), the Crystal family of methodologies (Cockburn 2002), Feature Driven Development (FDD) (Palmer and Felsing 2002), the Rational Unified Process (RUP) (Kruchten and Royce 1996), Dynamic Systems Development Method (DSDM) (Stapleton 1997) and Adaptive Software Development (ASD) (Highsmith 2000). Individual agile development methodologies can vary significantly in implementation. XP for example, has a strong focus on the technical aspects of software development, while Scrum attempts to build agility into the project management approach.

Numerous studies have reported a broad range of benefits from agile software development, including a reduction in risk (Da Silva and Da Cunha 2006, Abrahamsson et al. 2002), a reduction in code defect density (Fitzgerald, Hartnett and Conboy 2006), a decrease in the number of change requests (Da Silva and Da Cunha 2006), increased customer satisfaction (Da Silva and Da Cunha 2006), increased productivity and time-to-market (Reifer 2002, Kettunen et al. 2010), improved social dynamics (Whitworth and Biddle 2007, Syed-Abdullah, Holcombe and Gheorge 2006), and improved software development educational outcomes (Williams and

Kessler 2000, Van de Grift 2004). However, despite this body of research concerning the benefits of agile software methods, it has also been noted that much additional research is required in order to fully evaluate the impact of agile methods (Dyba and Dingsoyr 2008).

Agile software development methods may also suffer from a number of limitations, including that they are difficult to scale to large software development settings (Constantine 2001, Cockburn and Highsmith 2001), that they require *premium* people (Constantine 2001, DeMarco and Boehm 2002) (especially for certain practices (Boehm 2002) such as refactoring (Fowler et al. 1999)), that they place impractical demands on customer collaboration (Greer and Conradi 2009), and that they lack an upfront design (DeMarco and Boehm 2002). Other research has also cautioned that the *a la carte* approach to implementing agile methods, whereby some practices are included and others are disregarded, can reduce the effectiveness of agile development (Bowers, Sangwan and Neill 2007). It has also been suggested that agile approaches may actually be plan driven – that they aggressively plan for change (Wang, O'Conchuir and Vidgen 2008), something that perhaps traditional approaches could facilitate without the need for agile methods (Hansson et al. 2006). Furthermore, in common with process maturity frameworks and quality management standards, agile methods have also been criticised for lacking a continuous improvement dimension (Visconti and Cook 2004).

As demonstrated in the preceding paragraphs and as is the case for more traditional approaches such as process maturity frameworks and quality management standards, opinion is divided regarding the impact of agile development methods. Different research contributions have highlighted numerous benefits and limitations to both agile development methods and traditional development approaches. The types of benefits that have been reported for the different approaches span a broad spectrum: from improvements in the predictability of schedule and budgetary commitments, to increased flexibility to respond to changing requirements, to improved quality of products. Equally, the reported limitations are broad, ranging from additional cost, to increased dependence on team members, to decreased productivity. Since flexibility may be more important than cost in one setting, while in a different setting the cost may be a greater concern than the quality, it is vitally important that the software development process should be harmonised with the needs of individual software development contexts.

2.3.8 The importance of situational context

The basic requirement of a software development process is that it "*should fit the needs of the project*" (Feiler and Humphrey 1992 p6). Since software development projects and companies vary widely, it is not surprising to discover that no single prescribed software development approach "*is universally deployed or even universally useful*" (Jones 2008 p13). As a result, it has been reported that some projects will simply not benefit from being implemented in an agile fashion (Hawrysh and Ruprecht 2000, Glass 2001), while other research suggests that agile development can address some of the issues associated with traditional approaches (Petersen and Wohlin 2010).

Many researchers and practitioners have acknowledged the important role of situational context, commenting that "*the most suitable method is contingent on the context*" (Benediktsson, Dalcher and Thorbergsson 2006 p97), that "*managers must… evaluate a wide range of contextual factors before deciding on the most appropriate process to adopt for any given project*" (MacCormack and Verganti 2003 p230), that the chosen development approach should "*best fit the conditions, product, talent, and goals of the markets and organisations*" (Subramanian et al. 2009 p118), and that process improvement initiatives should be "*adjusted to the particular situation and … should not slavishly follow one of the comprehensive approaches*" (Kautz 1998 p223). Indeed, some of the most influential contributors in software development over the past 30+ years have explicitly acknowledged the critical role of situational context in guiding the software development process (Boehm and Turner 2003, Glass 1996). Furthermore, the absence of situational context descriptions in published SPI-related studies has been identified as an area of weakness (Unterkalmsteiner et al. 2012).

Despite the acknowledged critical role of situational context when designing and maintaining a software development process, it is surprising that there has not been a concerted effort to identify and classify the factors of the situational context that can influence software development process decisions. As noted in the opening chapter, the third research question associated with this thesis seeks to determine *what aspects of situational context are commonly experiencing change in software SMEs.* Since no comprehensive framework of the situational factors affecting the software development process exists at present, it is incumbent on this research to create one.

Given the rich diversity of situational contexts for software development projects, it would be a mistake to claim that one particular software development process is suited to all software development settings (Boehm and Turner 2003). Hence, a number of earlier research efforts have suggested that for each software development setting, there is a need to synthesise aspects of various different process approaches (Boehm 2002, Subramanian et al. 2009, de Cesare et al. 2010, Lycett et al. 2003, Ruparelia 2010). Furthermore, evidence of the benefits of such process synthesis has been established in earlier studies (Karlstrom and Runeson 2006, Pikkarainen et al. 2008). Therefore, it is likely that in practice, software development processes are constituted by an assortment of components from the different prescribed software development approaches (including CMMI, ISO/IEC 15504, ISO 9000 and agile methods). Consequently, a thorough investigation of SPI in practice, such as that proposed in this research, would benefit from focusing on changes to the process components rather than framing the research in the context of any single development approach.

A final important consideration relates to the volatility of situational contexts. In addition to the need to find an optimal process for a given environment, "*it is [also] reasonable to assume that the optimal process is not static but is organization-dependent and time-dependent, and will have to be modified as the context in which the organization operates evolves*" (Poulin 2007 p25). We should therefore expect process adaptation to be a regular feature of a fully functioning software development environment. However, as noted in the earlier sections, many of the dominant software development process approaches – including ISO/IEC 15504, ISO 9000 and agile development methods – have been criticised for failing to address the basic and undeniable recurring need for process adaptation (Jarvinen, Hamann and Van Solingen 1999, Coallier 1994, Grady 1997, Visconti and Cook 2004).

2.3.9 Summary

This section examined the role of the software development process and SPI. The process maturity and capability concepts were discussed and the CMMI and ISO/IEC 15504 process maturity frameworks were presented and appraised. The ISO 9000 quality standard was also introduced and assessed, as were the more recent agile software development approaches. This section has also highlighted that although there are a number of dominant prescribed approaches to software development, there is no one-size-fits-all when it comes to the software development process. Individual situational contexts, that are themselves under constant change, inform decisions regarding the optimisation of the software development process for any given environment. Smaller software development organisations may be more exposed to changing situational circumstances than larger, more stable organisations. In the following section, a brief review of the software SME sector is presented.

2.4 Software SMEs

In this section, the software SME sector is examined in terms of the number and proportion of individual organisations that qualify as SMEs. The term SME refers to a genre of company that is essentially not a large organisation. There are a number of reasons for conducting this research in the SME sector. Firstly, as illustrated in the following paragraphs, within the EU most companies are SMEs and most people are employed by SMEs; and within the software development industry, a high proportion of companies are SMEs. Therefore, the research findings have the potential to be applicable to a large number of companies. Secondly, Ireland is the primary setting for the research and as the evidence in the following paragraphs will demonstrate, the significant majority of Irish-based software development enterprises are SMEs. Finally, small companies are noted for their ability to quickly adapt to emerging challenges and therefore provide a fertile environment for the study of the impact of SPI.

According to the European Commission, the category of micro, small and medium-sized enterprises (SMEs) is made up of "*enterprises which employ fewer than 250 persons and which have an annual turnover not exceeding EUR 50 million, and/or an annual balance sheet total not exceeding EUR 43 million*" (European Commission

2003 p37). There are two further classifications within the SME category: small and micro enterprises. A small enterprise is defined as employing "*fewer than 50 persons and whose annual turnover does not exceed 10 million euro*" while a micro enterprise is defined as employing "*fewer than 10 persons and whose annual turnover does not exceed 2 million euro*" (European Commission 2003 p4).

Within the European Union (EU), 98% of an estimated 19.3 million enterprises fall under the European Commission definition of an SME category (Lukacs 2005). Furthermore, SMEs account for approximately 66% of total employment within the EU (Lukacs 2005). These general business statistics appear to be broadly applicable to the software development industry, where a high proportion of software development companies are SMEs (Fayad, Laitinen and Ward 2000, Richardson and von Wangenheim 2007, Pino, Garcia and Piattini 2008, Valtanen and Ahonen 2008). Therefore, by conducting this research in the SMEs sector, there is potential to deliver benefits to a wide community.

Ireland is the primary setting for the research and the statistics show that there is a pool of suitable organisations available for a field study such as that required by this research. According to Enterprise Ireland (EI), the Irish governmental development body for indigenous companies, Irish software businesses alone employ over 10,000 people in more than 500 individual companies (Enterprise Ireland 2009). The significant majority, 98.1%, of such software companies have been shown to comprise 100 people or less (Crone 2002). Moreover, EI (2009) sets out its strategy for the development of the indigenous software industry, stating that it will "*work with its clients to help them grow sales, so that a significant number of companies will have sales in excess of €20 million by 2013*". Even if this aspiration is successfully accomplished, many of the organisations will likely remain in the SME category.

In addition to their proliferation, one of the main competitive advantages of software producing SMEs is their ability to aggressively adapt and quickly deliver new products; this is especially true in software companies that are in an early state of development (Sutton 2000). With rapidly changing environments in software SMEs, there is a need for continual process adaption – and therefore, SMEs provide an ideal setting for the examination of the interplay between SPI, situational change and business success.

2.5 SPI in software SMEs

In section 2.3, many of the advantages and disadvantages of established approaches to software development were discussed, revealing that there are many direct and indirect benefits from SPI (Zahran 1998). Such benefits include increases in productivity, product quality and customer satisfaction, improvements to budget and schedule adherence and decreases in costs, cycle times and process complexity. However, it has been observed that the benefits of SPI can be difficult to measure (Rozum 1993, Mathiassen, Ngwenyama and Aaen 2005). In one attempt to financially quantify the benefits of SPI, Rico (2004) presents an approach for comparing the investment with the return, or in other words calculating the Return On Investment (ROI). However, in practice, ROI is inconsistently calculated, which has resulted in confusion and general scepticism (Erdogmus, Favaro and Strigel 2004). In spite of this, some research has investigated the ROI associated with SPI, with Van Solingen (2004) presenting a review of several such studies that were carried out in large development organisations – determining that the average ROI for SPI is 7:1, i.e. for every one dollar invested, seven dollars are returned.

Much of the literature of SPI-related financial ROI is centred on studies in large organisations – as demonstrated by the company listing presented by Van Solingen (2004). While comparable information for small software development companies is less evident, they can derive benefits from SPI (Kautz 1998). Sanders (1998) examines the benefits accruing to small software development organisations from SPI: one organisation, *Cunav*, significantly reduces the rework effort by improving the requirements capture process while a second organisation, *Peregrine*, increases overall capability by investing in source code and defect management processes. Another study of SPI in small organisations concludes that it is possible to implement software processes in a beneficial and cost-efficient manner considering their specific business goals, characteristics, and resource limitations (Von Wangenheim et al. 2006). While Sanders (1998) and Von Wangenheim et al. (2006) demonstrate that SPI can make a significant contribution to improving software development in small companies, there is no attempt to examine the association between SPI and the successful achievement of business goals.

In a related study by Cater-Steel and Rout (2008), process improvement is reported to have a positive long-term effect on businesses. However, by focusing just on traditional views of business success, such as financial and headcount measures, they lack a comprehensive examination of the broader business success considerations for software development organisations (such as those identified in the Holistic Scorecard (HSC) (Sureshchandar and Leisten 2005)). Furthermore, they also lack the type of multi-phased business success examination that is necessary to minimise the effect of false or biased recollection on the part of interviewees (a point that is discussed in more detail in chapter 6).

Other studies also demonstrate the benefits of SPI in small organisations. Ferreira et al. (2007) show that *BL Informatica* successfully grows its headcount through the successive implementation of quality management standards and process maturity reference models, including ISO 9000 and CMMI. However, as previously identified, these approaches may not be well suited to the needs of small companies in general (Miluk 2005, El Emam and Birk 2000) and evidence suggests that SMEs have not widely adopted such approaches (Coleman and O'Connor 2008, McConnell 2002). Moreover, the work presented in Ferreira et al. (2007) does not attempt to correlate SPI actions with business success.

Fleck (2004) proposes that requirements documentation, change control and communication are among the most important process activities – but the study, which presents a light-weight development process for very small organisations, is carried out in just one company. In other research focused on a single SME, Biro et al. (2000) find that, owing to SPI initiatives, *MemoLuX* is able to reduce production costs while at the same time generating increased business opportunities. Further evidence of the benefits of a light-weight SPI approach to a single SME are presented in Scott et al. (2002) where improvements in project estimation and customer relations are observed. However, along with Fleck (2004) and Biro et al. (2000), Scott et al. (2002) does not attempt to examine the relationship between SPI and the broad spectrum of possible business objectives.

Deephouse et al. (1995) use a multiple-company survey to investigate the effectiveness of individual process activities, finding that effective planning and cross-functional teams are regarded as being of major importance to project outcomes. The respondents are described as being experienced software engineers from a range of large and small companies, meaning that Deephouse et al. (1995) lack a specific SME focus. In addition, there is no attempt to investigate the influence of SPI on the achievement of business goals – but rather on the successful outcome of a specific project.

While the benefits of SPI to SMEs have been demonstrated through studies such as those in the preceding paragraphs, Niazi (2006) concludes that much more evidence in favour of SPI for SMEs is required in order to justify a commitment to SPI programmes. In more recent works, Niazi et al. (2008) and Niazi and Babar (2009) attempt to establish the perceived value of the specific practices of the following CMMI Level 2 process areas: *requirements management*, *process and product quality assurance* and *configuration management*. These studies acknowledge the need for finer granularity in relation to the perceived value of different software processes and SPI practices. However, the restriction in study implementation to just a subset of the CMMI Level 2 practices is problematic, since it has been claimed that such practices are not necessarily of primary benefit to small companies (Von Wangenheim et al. 2006). Consequently, the framework underlying the studies carried out by Niazi, Babar and Ibrahim (2008) and Niazi and Babar (2009) may not represent the most appropriate point of departure for research into the key practices for small software development companies. Moreover, Niazi, Babar and Ibrahim (2008) and Niazi and Babar (2009) do not attempt to examine the relationship between SPI and business success.

2.6 Core research focus

Up to this point, this chapter has gradually developed a broad view of the software development process and SPI, with the more recent sections examining the role of SPI in software SMEs. Earlier research has focused on the benefits of SPI, such as reducing the amount of rework and improving project estimation. However, to date no published research has investigated the relationship between SPI and a broad spectrum of business success considerations. Therefore, this research will be designed from the outset to examine in detail if SPI activity is positively associated with business success, with the key focus being the software SME sector.

While members of the software process and SPI communities believe that SPI does contribute to business success, there is only weak or indirect evidence to support this position at present. As a result, it is reported in recent studies that software SMEs can have a low software development process priority (Baddoo and Hall 2003), choosing only to implement SPI in response to negative business events (Coleman and O'Connor 2008). This researcher believes that it is important to examine the relationship between SPI and business success, and specifically, to investigate whether or not SPI is positively associated with business success in software SMEs. Were evidence of a positive relationship between SPI and business success to be discovered, the motivation for SPI in software SMEs would be greatly improved. Evidence of such a nature would highlight the importance of maintaining an SPI focus in software SMEs, not just for localised improvements in quality or other specific criteria, but for support of business success in the broader sense.

When examining the relationship between SPI and business success, numerous different approaches could be adopted. Since no single software development method is universally implemented or even universally useful (Constantine 2001), an important prerequisite for an examination of SPI is that it should address a comprehensive spectrum of the atomic software development process components – rather than just pursuing the process constructs set out in a specific software development approach. In this respect, this research has elected to examine the amount of SPI activity in software SMEs using ISO/IEC 12207 as the underlying process reference framework. Since ISO/IEC 12207 is designed to identify all of the atomic components of a software development process (and not the actual organisation of the process components), it presents as an ideal framework for examining SPI independent of any particular software development approach. Chapter 4 of this thesis explains why a new approach for examining SPI activity in software SMEs was developed for this study, along with a detailed description of the approach itself.

As indicated in the earlier sections of this chapter, when it comes to process optimisation, the most basic of principle is that the process should fit the situational context. Although the situational context is continually changing, it does so at different rates and in different ways in each organisation. Therefore, the amount and type of SPI required for process optimisation in one organisation could be significantly different to the amount and type of SPI required in a different organisation. For this reason, any evaluation of the relationship between SPI and business success would be incomplete if it did not also consider the degree of situational change. Although situational context is generally acknowledged as being an important consideration for an optimal software development process (Boehm and Turner 2003, Benediktsson, Dalcher and Thorbergsson 2006, MacCormack and Verganti 2003, Subramanian et al. 2009, Kautz 1998, Glass 1996), a review of the literature did not produce a comprehensive reference framework of the situational factors affecting the software development process. Therefore, this research will systematically develop a reference framework of the situational factors affecting the software development process, the details of which are outlined in chapter 5.

With respect to examining the extent of business success, it is important to incorporate a comprehensive list of the various success considerations for software companies. Since earlier SPI studies were not designed to examine the broad spectrum of business success considerations, there is little guidance on how this aspect of the research should be conducted. As is described in detail in chapter 6, this research adopts the HSC (Sureshchandar and Leisten 2005) to achieve this purpose, and furthermore, a novel two-phased approach to robustly examining business success is developed. This new approach presents a thorough and reliable method for making determinations regarding business success in software development organisations.

In conclusion, there are three central themes to the investigations to be carried out in this research: the first, to examine the amount of SPI activity; the second, to determine the extent of business success; and the third, to identify the degree of situational change. Using these three distinct modes of inquiry, this research can make a significant addition to the body of software process and SPI knowledge, not just concerning the relationship between SPI and business success in software SMEs, but also regarding the way that researchers and practitioners view SPI. There is no one-size-fits-all when it comes to the software development process, and research suggests that in practice software development organisations pick and choose aspects of different prescribed approaches in order to best satisfy their needs. Therefore, as a research community we should focus our energies on trying to maximise the opportunities provided by the unique set of characteristics in each software development setting – and this study represents a step in that direction.

2.7 Summary

This chapter started by outlining the approach to the literature review associated with this research, followed by a general review of the software process and SPI landscape. The specific literature relating to SPI in software SMEs was also discussed. Finally, this chapter established the novelty and value of examining the relationship between SPI and business success in software SMEs. The following section, Part 2, outlines the methodology adopted in order to investigate the relationship between the three principle phenomena under investigation in this study: SPI, situational change and business success in software SMEs.

Part 2 Research Methodology

The second part of the thesis contains chapter 3, which is dedicated to the study methodology. Initially, the established research paradigms are presented, followed by an explanation of the suitability of mixed method research in this study. The approach to generating data for the study is also presented.

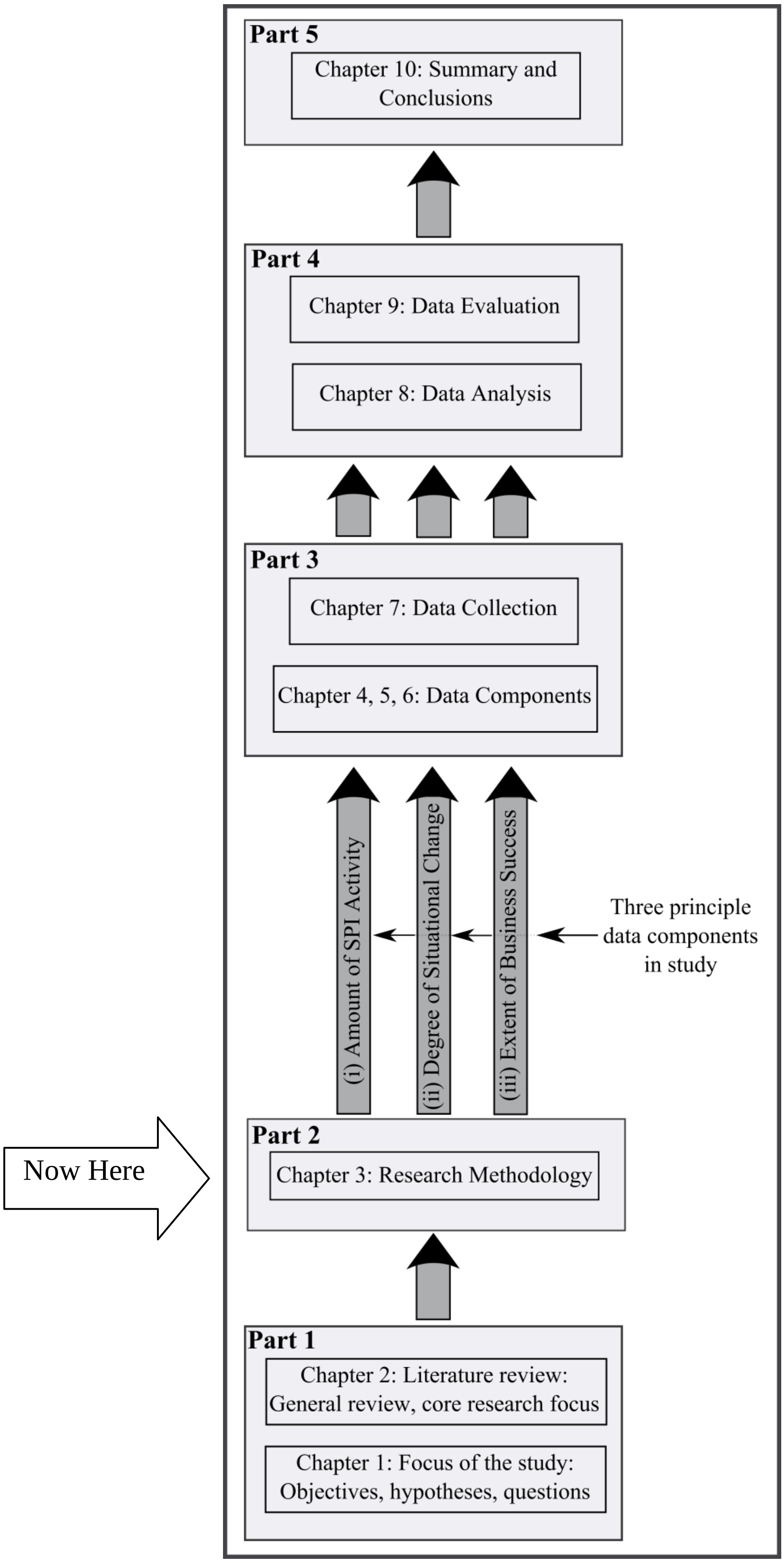


Figure 5 Map of Thesis – Part 2

3 Research Methods and Study Methodology

3.1 Introduction

This chapter presents a review of the principal philosophical approaches to research. The predominant research methods and methodologies are presented and reviewed in terms of their suitability for the present research, after which the data generation process is identified. Finally, there is a discussion on the extent to which cause-and-effect can be determined in studies of this type.

3.2 Research paradigm

Our beliefs in relation to the workings of the world (cosmology), our understanding of human behaviour (ontology), how we acquire knowledge (epistemology), and our views in relation to the conduct of people and society (values) are not necessarily conscious – but they have an impact on and are affected by how we relate to others and to the world (Peile and McCouat 1997). This constellation of beliefs, techniques, values, and actions can be described as a paradigmatic position (Haworth 1984), with Oates (2006) explaining that paradigms may be broadly shared among a group or community. Paradigms, and the specific methods used by them, are therefore based on various philosophical foundations and concepts of reality (Peile and McCouat 1997). Classically, two main research paradigms exist: *positivism* and *interpretivism*, and these are traditionally presented as being fundamentally opposed (Creswell and Plano Clark 2007).

3.2.1 Positivism

Positivism is concerned with the notion of a singular reality, that there exists just one truth which can be discovered by objective and value-free inquiry (Feilzer 2010). Positivist research can be generally characterized as theory testing research (Myers 1997) - it is deductive in nature, starting with a theory based on a conceptual world and then testing the theory empirically in the real word (Alqatawna et al. 2009). It perceives the world as fixed, with measurable phenomena that can be objectively and repeatedly observed and investigated with structured instrumentation independently from the researcher (Chua 1986).

3.2.2 Interpretivism

Interpretivism emphasises the role of people and how they interact with and perceive the phenomenon under investigation (Alqatawna et al. 2009). The intention of interpretivist research is to increase the depth of understanding of the phenomenon under investigation by taking into consideration the cultural and contextual situation (Chua 1986). Interpretivism doesn't seek to test a hypothesis but rather it aims to create a holistic understanding of a phenomenon by exploring how all the factors in the social context are related and interdependent (Oates 2006). With interpretivist research, the researcher is immersed in the phenomenon under investigation in an effort to determine the socially constructed reality (Firestone 1987).

3.3 Research methods and methodologies

Although they are sometimes used interchangeably, as in Tashakkori and Teddlie (1998), the terms *research method* and *research methodology* are considered by some to be distinct, with Mingers (2001) offering the following definitions:

- Research methods are concerned with the approaches to or techniques of data collection (such as surveys and reviews).
- Research methodologies relate to the overall set of guidelines that support the research activities (in order to generate reliable and valid research results).

The family of methods associated with the positivist paradigm are classified as *Quantitative Methods* while the corresponding interpretivist family of methods are termed *Qualitative Methods* (Naslund 2002). Table 1 presents a taxonomy (Galliers 1991) of the more common research methods, indicating their association with the positivist and/or interpretivist paradigms.

While some debate exists in relation to the efficacy of combining quantitative methods and qualitative methods within the scope of individual research endeavours (Sale, Lohfeld and Brazil 2002), in practice research projects do incorporate aspects of both approaches (Tashakkori and Teddlie 2003), giving rise to the concept of *mixed method* research. Indeed, mixed method research has itself been presented as a research approach that is distinct from the positivist and interpretivist paradigms, with some considering it the third major research paradigm (Johnson and Onwuegbuzie

2004). The following sections provide an overview of quantitative, qualitative and mixed research methods.

Positivist	**Interpretivist**
Lab Experiments	Subjective/Argumentative
Field Experiments	Reviews
Surveys	Action Research
Case Studies	Case Studies
Theorem Proof	Descriptive/Interpretive
Forecasting	Future Research
Simulation	Role/Game Playing.

Table 1 Taxonomy of research methods (Galliers 1991)

3.3.1 Quantitative research methods

Quantitative research methods are systematic, scientific investigations that use numerical representation and the manipulation of observations for the purpose of describing and explaining the phenomena that those observations reflect (Babbie 2009). Therefore, quantitative research methods involve empirical research where the data is informed using numbers (Punch 1998) that are collected in a form that is suited for statistical analysis, and that is intended to be objective, non-reactive and representative (Creswell 2003). Myers (1997) explains that the goal of quantitative methods is to determine whether the generalisation of a theory holds true, with Bryman (2007) expressing that such a goal is served by the deductive nature of quantitative research methods.

Also known as hypothetico-deductive methods, quantitative research methods consist of the following components (Jankowicz 1995):

- A formally expressed general statement which has the potential to explain things (Theory)
- A deduction that, if the theory is true, then you would expect to find a relationship between at least two variables, A and B (Hypothesis)
- A careful definition of exactly what you need to measure so as to observe the variances in A and B (Operational Definition)

- The carrying out of the observations (Measurement)
- The drawing of conclusions about the hypothesis (Testing)
- The drawing of implications back to the theory (Verification)

According to Creswell (2003), the design of a quantitative research study can be classified under one of three categories: *Experimental*, *Quasi-Experimental* and, *Descriptive and Correlational*. *Experimental* and *Quasi-experimental* designs are used to study cause-and-effect, while *Descriptive and Correlational* designs study the research variables in their native environment. The following five techniques can be utilised in order to collect quantitative research data: Social surveys, Experiments, Analysis, Structured observation, and Content analysis.

3.3.1.1 Suitability of quantitative research methods for present study

Social surveys can be used to obtain data from a large sample that is known to be representative of a broader population, sometimes taking the form of a focused interview that extracts specific information. Since this research is concerned with collecting data from multiple participating organisations, aspects of the social survey technique are incorporated to some extent in the present study, the details of which are provided later in this chapter.

Experiments can be used to stimulate a *study group*, while a *control group* that is not subjected to the stimulus is also maintained. They are used to "*test the impact of a treatment on an outcome, controlling for all other factors that might influence that outcome*" (Creswell 2003 p.154). The present study is carried out in an environment where it is not possible to definitively control for all factors – since a complex cocktail of factors other than SPI have an influence on business success. Furthermore, this research is not concerned with stimulating a study group, it is observational and non-interventionist. Therefore, experimentation is not a useful research technique for this study.

Analysis can be performed on pre-existing data sets. However, the type of information that is required in the present study (such as business objectives, SPI data and situational data) is not necessarily stored in pre-existing data sets and consequently, analysis is an inappropriate technique for this study.

Structured observation can extract focused data in a controlled and pre-determined fashion. As the name suggests, the process involves observing behaviour and actions. This form of research would not yield the type of information required in this study – since some of the data items (such as the business objectives) are not necessarily directly evident in behaviours and actions (or even in discussions); and even if all of the data could be collected using just structured observation, the observation time requirement would be very significant and consequently, the cost would be prohibitive.

Defined as "*any technique for making inferences by objectively and systematically identifying specified characteristics of messages*" (Holsti 1969 p.14), *Content analysis* can be used to extract data from mass-media products. However, the present study does not lend itself to extracting information from mass-media data, as there is no pre-existing data suite from which the phenomena under investigation can be reliably observed. Therefore, content analysis is not considered to be an appropriate research technique.

Quantitative research methods involve the collection and analysis of data in a numeric or symbolic form and are essentially concerned with the establishment of indisputable facts (Blaxter, Hughes and Tight 2001). Such methods are most effective when the content is sufficiently controlled so as to ensure that the study events are free of undefined influence, under which circumstances the reproducibility is high and the results are likely to reliably predict the outcome of the same event in future (Lakshman et al. 2000). However, a major disadvantage of quantitative research is that in general, many of the social and cultural aspects are lost or are treated in a superficial manner (Myers 2009). This research is concerned with collecting data in relation to three phenomena (SPI activity, situational change and business success) that are not easily subjected to quantitative measurement. Therefore, although the social survey aspect of quantitative research offers some possibilities for inclusion as a research data collection technique, in general, quantitative research methods are not well harmonised with the data collection needs of this research.

While qualitative research methods do not offer the generalisability and precision of their quantitative counterparts (Lakshman et al. 2000), they do provide a framework under which human behaviour, attitudes and beliefs can be examined in depth. The following section examines the suitability of qualitative research data collection techniques for this study.

3.3.2 Qualitative research methods

According to Creswell (2003), qualitative research takes place in a natural setting, uses methods that are interactive and humanistic, is emergent rather than tightly pre-configured and views social phenomena in a holistic fashion. It involves the non-numerical examination and interpretation of observations in order to discover underlying meanings and patterns of relationships (Babbie 2009). Therefore, qualitative research methods aim to establish detailed knowledge regarding human, social and cultural phenomena – a focus which may come at a cost to the generalisability of the findings (Patton 1990). In contrast to quantitative research, qualitative research focuses on exploring information in depth rather than in breadth (Blaxter, Hughes and Tight 2001), developing hypotheses through the use of largely inductive research methods (Bryman 2004).

The following primary qualitative research methodologies can be employed in a qualitative research project (this list excludes surveys, which can be implemented in both quantitative and qualitative research): Phenomenology, Ethnography, Case Studies, Action Research, and Grounded Theory.

3.3.2.1 Suitability of qualitative research methods for present study

Phenomenological research is concerned with the identification of the essence of the human experience of phenomena, as described by the participants in a study (Creswell 2003). Since it seeks to fully understand the personal experience of phenomena, phenomenology is sometimes considered to be a philosophy as well as a methodology. As will be outlined later, the present study incorporates aspects of the phenomenological methodology.

Ethnographic research is concerned with generating and analysing qualitative data that has its origin in first-hand experience, with the primary ethnographic method being participant observation (Schwandt 2007). Other forms of ethnographic research include interviewing, the collection and analysis of artifacts, and the collection of oral records. Ethnography is generally characterised by the ethnographer spending time among the research population, observing and recording data over this period of time (Taylor 2010). While ethnography could be employed to some extent in the present study, it does not represent an effective or efficient means of obtaining the required data. It is not necessary to spend extended periods of time with the research population in order to determine the extent of business success or the amount of SPI activity. Therefore, an ethnographic type research engagement is not considered suitable as a research data collection technique.

Case Studies involve the in-depth research of a program, an event, an activity, a process or one or more individuals (Creswell 2003). Although the case study takes place over a sustained period of time, the case is time-bound, and the researchers are permitted to deploy a variety of data collection procedures in the pursuit of detailed information. It has been noted by Verschuren (2003) that it is also possible for case studies to be utilised in quantitative research. Case studies are used when the research is concerned with insight and interpretation rather than hypothesis testing (Merriam 1998), and in studies where *how* and *why* questions are asked (Yin 1994). Since the present study is concerned with examining *what* type research questions (refer to chapter 1), the case study method is considered to be an inappropriate research data collection technique.

The term *Action Research* was coined by social psychologist Kurt Lewin (1890-1947) in the 1940s to describe a type of research that united the experimental approach of social science with programs of social action to address social problems (Verschuren 2003). The focus includes not just the analysis of practice but also its improvement, and therefore, action research is considered to be interventionist in nature. Action research is a *learning by doing* process which involves problem identification, problem solving, solution evaluation, and the entire process may be repeated if the desired outcomes are not achieved (O'Brien 1998). The present study is concerned with establishing if SPI is positively associated with business success in practice, and not with improvement through intervention. Consequently, action research is not a suitable research approach for the present study.

Although the term *Grounded Theory* can be used in a non-specific way to refer to any approach to developing theoretical ideas that begins with data, grounded theory methodology is a specific, highly developed, rigorous set of procedures for producing formal substantive theories of social phenomena (Verschuren 2003). This involves the analysis of qualitative data by simultaneously employing deductive, inductive and verification techniques in order to develop theory. In this study, the type of research data, the research hypotheses and research questions are established early in the study. This study is not aimed at discovery in the deductive sense or at theory development – but rather at hypothesis testing and answering research questions. Therefore, grounded theory does not represent an appropriate research method.

Qualitative research methods present an alternative research approach to quantitative research methods and enable the in-depth context-sensitive examination of complex social phenomena. It has also been noted that "*in many situations, researchers can put together insights and procedures from both approaches to provide a superior product*" (Johnson and Onwuegbuzie 2004 p.17), a practice that is known as *mixed method research*.

3.3.3 Mixed method research

There exists some confusion as to what exactly constitutes mixed method research, with one possible definition stating that it involves the collection or analysis of both quantitative and/or qualitative data in a single study in which the data are collected concurrently or sequentially, are given a priority, and involve the integration of the data at one or more stages in the process of research (Creswell et al. 2003). More concisely, mixed method research can be considered as "*research that combines qualitative and quantitative methods to collect, analyze, and present both types of data*" (Petter and Gallivan 2004 p.2).

Mixed method research is an expansive and creative form of research that is inclusive, pluralistic and complementary, allowing researchers to adopt an eclectic approach to method selection and the thinking and conduct of research (Johnson and Onwuegbuzie 2004). The philosophy most commonly associated with mixed methods research is *pragmatism* (Teddlie and Tashakkori 2009), which offers an alternative world view to positivism and interpretivism and which focuses on the problem to be researched and the consequences of the research (Creswell and Plano Clark 2007, Brewer and Hunter 1989, Miller 2006). Since it accepts that there are singular and multiple realities that are open to investigation, pragmatism sidesteps the contentious issues of truth and reality that are extant in positivist and interpretivist debates (Feilzer 2010, Dewey 1925, Rorty 1999).

Pragmatists hold an *anti-representational* view of knowledge arguing that the aim of research isn't necessarily to most accurately represent reality, but to be useful, to aim at utility (Rorty 1999). Pragmatism does not prescribe or preclude any given method or combination of methods and it does not expect to find fixed causal links or truths but rather, it aims to interrogate a particular question or phenomenon with the most appropriate research method (Feilzer 2010). This philosophy of designing the research methodology around the study is a cornerstone of pragmatism, where finding the closest possible match of theory and method is the paramount criterion for judging "*the legitimacy for a method*" (Hanson 2008 p.107).

Under a pragmatic worldview, a key starting point in the research method selection is the definition of "*what you want to find out*", which in turn "*leads inexorably to the question of how you will get that information*" (Miles and Huberman 1984 p.42). Therefore, the research questions are the embodiment of the research purpose, and they inform the method and methodological selection (Creswell and Plano Clark 2007, Bryman 2007, Brewer and Hunter 2005, Krathwohl 2004, Newman and Benz 1998).

As indicated in the earlier chapters, this thesis investigates the relationship between SPI and business success. This involves a field-based component that determines the amount of SPI activity and the extent of business success, as well as a correlation of these two phenomena. Viewed pragmatically, this necessitates the use of both qualitative and quantitative techniques, and is therefore suited to a mixed method study. Specifically, and as is outlined in more detail later, this research merges aspects of social survey data collection (quantitative) and phenomenology (qualitative) in the form of in-depth interviews.

3.3.3.1 Using mixed method research in present study

Prior to outlining the research method in detail, it is worth examining the motives for adopting a mixed method approach. According to Greene, Caracelli and Graham (1989), there exist five separate motives for carrying out mixed method research, as follows:

- *Triangulation*: the collection and analysis of different types of data in order to improve the precision of the results (Jick 1979).
- *Complementarity*: the use of mixed methods in order to better understand a phenomenon and to enhance the detail and depth of information obtained (also known as *elaboration* (Greene, Caracelli and Graham 1989, Rossman and Wilson 1985, Rossman and Wilson 1994)).
- *Development*: also known as *facilitation* under Hammersley's (1996) classification, involves the use of results derived from one research method to assist in the primary study which uses an alternative research method.
- *Initiation*: the discovery of contradictions in data so as to support the development of a new understanding of problems (Rossman and Wilson 1985).
- *Expansion*: in order to build a more comprehensive understanding or solution for a problem, multiple phenomena can be investigated using mixed methods to broaden the scope and breadth of the study.

The use of mixed method research in the present study is influenced by two of these motives: *Triangulation* and *Expansion*.

This study will involve the collection of data from practice, with this data taking the form of both numbers and percentages (in the case of business targets and their achievement), and expert opinions (in the case of the amount of SPI activity and the degree of situational change). Using *Triangulation*, these different types of data are collected and evaluated together in order to examine the research hypotheses and answer the research questions.

The research is also motivated by a need for *Expansion*. According to Greene, Caracelli and Graham (1989), *Expansion* seeks to extend the breath and range of inquiry by using different methods for different inquiry components. This is an important consideration for the present study since different frameworks are required for the examination of SPI activity, situational change and business success. Therefore, in order to facilitate the broad nature of the inquiry, it is necessary to expand the inquiry methods such that they are suitable for the given inquiry task.

While this research can be considered to be motivated by two of the five motivations in the Greene, Caracelli and Graham (1989) scheme, it is worth noting that the scheme itself may be a limiting classification – "*in that it boils down the possible reasons for conducting multi-strategy research to just five reasons*" (Bryman 2006 p.105). Byrman (2006) suggests that in practice, as many as seventeen different motivations may exist. Of these seventeen motivations, at least six can be considered pertinent in the decision to adopt a mixed method approach in this research: *Offset, Completeness, Instrument Development, Credibility, Utility* and *Context.*

Bryman (2006) offers the following explanations for these six motivations: *Offset* refers to the suggestion that the research methods associated with both quantitative and qualitative research have their own strengths and weaknesses so that combining them allows the researcher to offset their weaknesses to draw on the strengths of both, while *Completeness* refers to the notion that the researcher can bring together a more comprehensive account of the area of inquiry in which he or she is interested if both quantitative and qualitative research are employed. This study appeals to both of these motivations. Furthermore, this research can make a strong argument for *Instrument development*, which refers to situations in which qualitative research is employed to develop questionnaire and scale items – for example, so that better wording or more comprehensive answers can be generated. In addition, the research can claim to offer greater *Credibility* through the use of a mixed method approach and since combining quantitative and qualitative approaches renders a suitable approach for examining the key data items of interest, the *Utility* of the research is improved in terms of the utility of the findings for practitioners and researchers. Finally, *Context* refers to cases in which the combination is rationalised in terms of qualitative research providing contextual understanding coupled with either generalisable, externally valid findings or broad relationships among variables uncovered through a survey. This research makes use of surveys that collect data on a wide variety of variables and attempts to identify associations and relationships – therefore, the research is also motivated by *Context.*

As demonstrated in the preceding paragraphs, the present research is strongly motivated to incorporate a mixed method approach. Two of the recognised possible mixed method study motivations, as identified by Greene, Caracelli and Graham (1989), are evident in the present research needs. Furthermore, six of the seventeen motivations identified by Bryman (2006) are also evident. Therefore, it can be concluded that a mixed method study is appropriate in this case. The following section discusses the role of research questions.

3.4 Role of research questions

Punch (1986) and Bulmer (1988) have noted that although the delineation between research paradigms is clear at a philosophical level, when it comes to the choice of research method, and to research design concerns, this delineation breaks down. It is the research hypotheses and questions that serve as a guiding light for the research design, with Johnson and Onwuegbuzie (2004) stating that "*research methods should follow research questions in a way that offers the best chance to obtain useful answers*" (Johnson and Onwuegbuzie 2004 p.17-18). Bryman and Bell (2003) further assert that research questions are used to guide decisions about the "*research design*" and "*what data to collect and from whom*" (Bryman and Bell 2003 p.37). The research hypotheses have a similar function in terms of informing the decision regarding the research method. Tashakkori and Creswell (2007) observed that "*numerous scholars have reiterated the fact that research questions are shaped by the purpose of a study and in turn form the methods and the design of the investigation*" (Tashakkori and Creswell 2007 p.207), with Brewer and Hunter (2005), Bryman (2007), Creswell and Plano Clark (2007), Krathwohl (2004), and Newman and Benz (1998) all being cited in support of this statement. The research hypotheses and questions presented in chapter 1 inquire as to what aspects of the software process are undergoing SPI, what aspects of the situational context are changing, and what the meaning of business success is for software SMEs. These research hypotheses and questions demonstrate a need for a variety of different data types, and hence, a predisposition towards mixed-method interview-based techniques for data collection.

Easterby-Smith, Thorpe and Jackson (2008) suggest that interviews are an appropriate research method when: "*it is necessary to understand the constructs that the respondent uses as a basis for his or her opinions and beliefs about a particular matter*", when "*the step by step logic of a situation is not clear*", when "*the subject matter is highly confidential or commercially sensitive*", and when "*there are issues about which the interviewee may be reluctant to be truthful other than confidentially in a one-to-one situation*" (Easterby-Smith, Thorpe and Jackson 2008 p.145). While there is a possibility to extract the data using alternative techniques, "*the greatest value [of face to face interviews] lies in the depth and detail of information that can be secured… [and this] far exceeds the information secured from telephone and mail surveys*" (Cooper and Emory 1995 p.271). These observations from Cooper and Emory (1995), along with the conclusions of Easterby-Smith, Thorpe and Jackson (2008) indicate that interviews, both quantitatively and qualitatively constructed, offer the best possible approach to data generation for this study. Therefore, this research will utilise face-to-face interviews as the primary vehicle for data generation. The following section elaborates on the structure and composition of the research interviews.

3.5 Data generation for present study

As outlined earlier in this thesis, in order to conduct this research, three separate sets of data are required. Firstly, the amount of SPI activity is examined. Secondly, the degree of situational change is investigated. Thirdly, the extent of business success is determined. Each of these three items of data are suited to interview-based research but they need to be addressed separately, since they present some differences in terms of the data generation requirements. Prior to outlining the data generation method for each of these data components, it is useful to first review some concepts that are commonly used in relation to interviewing:

- *Structured Interview* – Also known as a standardised interview, a structured interview entails "*the administration of an interview schedule by an* *interviewer. The aim is for all interviewees to be given exactly the same context of questioning… [Meaning that] each respondent receives exactly the same interview stimulus as any other… Interviewers are supposed to read out questions exactly and in the same order as they are printed on the schedule*" (Bryman and Bell 2003 p.116).

- *Semi-Structured Interview* – The term semi-structured interview "*covers a wide range of instances. It typically refers to a context in which the interviewer has a series of questions that are in the general form of an interview schedule but is able to vary the sequence of the questions. The questions are somewhat more general in their frame of reference than that typically found in a structured interview schedule. Also, the interviewer usually has some latitude to ask further questions in response to what are seen as significant replies*" (Bryman and Bell 2003 p.119).
- *Open Question* – An interview question is described as open when "*respondents are asked a question and can reply however they wish*" (Bryman and Bell 2003 p.156).
- *Closed Question* – An interview question is described as closed when respondents are "*presented with a set of fixed alternatives from which they have to choose an appropriate answer*" (Bryman and Bell 2003 p.156).

Using the above descriptions of key interview concepts, suitable interview approaches are now developed for each of the three items of data required in this study.

3.5.1 SPI activity data

The software development team are responsible for the software development process. Therefore, it is possible to investigate SPI activity by interviewing members of the software development team. For example, closed questions such as "*have you made any modifications to your configuration management process?*" could be used to inquire about SPI activity. However, in order to determine the extent of the activity, further questions are required, such as "*To what extent have you modified your configuration management process?*" Therefore, in order to best address the research questions, it is necessary to employ a compound interview when examining the extent of SPI actions – and in view of the definitions of structured and unstructured interviews above, this approach is probably best described as a semi-structured interview employing largely closed questions.

3.5.2 Situational change data

Examining the degree of situational change will potentially require engagements with a variety of personnel in the participating organisations. Changes in relation to technology will be best addressed by technical personnel, while changes in relation to client management might be best addressed by senior management representatives. The questions will be largely similar in structure to the SPI activity questions, for example, "*To what extent has the volatility of customer requirements changed?*" Making determinations in relation to the degree of situational change therefore requires interviews that are semi-structured in nature and that contain a high proportion of closed questions.

3.5.3 Business success data

Business success concerns the examination of the extent to which the business is achieving its objectives. The executive team are generally responsible for the performance of the business and therefore, along with the business owner(s), represent a viable source for the collection of data in relation to the achievement of business goals. Predominately closed questions will be utilised in determining the business objectives, such as "*to what extent did you achieve your target for new client acquisitions?*" However, open questions discharged in a more semi-structured fashion will also be required so that the interview can support the disclosure of objectives that may be beyond the scope of the closed questions. Such questions will take the form of "*did you have any additional business objectives that have not been covered in the interview?*" Therefore, the business success interview is hybrid in nature, and appropriately so as it offers the best infrastructure to elicit the business success data requirements of the study. In view of the generally accepted definitions above, this hybrid type of interview is probably best described as a semi-structured interview comprising of mostly closed questions.

A summary of the research paradigm, data generation method, and data sources employed in generating the study data is presented in Figure 6.

	SPI Activity	Situational Change	Business Success
Data Source	Software Team	Across the Business	Senior Management
Paradigm	Interpretive & Positivist		
Methodology	Phenomenology & Social Survey		
Data Generation	Semi-Structured Interview		

Figure 6 Research paradigm, methodology and methods

3.6 Examining cause-and-effect

Having determined that a mixed method research design is required and that an interview-based data generation approach is appropriate, the next concern relates to the question of cause-and-effect. According to Lowstedt and Stjernberg (2006), "*the character and number of interviews are determined by the primary focus of the interview*" (Lowstedt and Stjernberg 2006 p.137). The primary focus of this research is to examine if SPI is positively associated with business success. However, cause-and-effect relationships in business success research are acknowledged as being less explicit than the type of causation found in experimentation, since meeting the "*ideal standard of causation would require that one variable always caused another and no other variable had the same causal effect*" (Cooper and Emory 1995 p.123). Given the complex nature of both businesses and software development, it is impractical to attempt to observe and measure everything that may account for the relationship between these two phenomena. Causal inferences must therefore be made, and although their permanency and universality cannot be guaranteed, they allow for the "*building of knowledge*", hence providing "*approximations to the truth*" (Cooper and Emory 1995 p.126).

In order to increase the generalisability and to improve the certainty of the findings, it is sometimes necessary to broaden the study beyond a single participant or organisation, that is, to increase the sample size – a recognised approach to resolving the question of certainty (Lee and Baskerville 2003). In this regard, a number of considerations are relevant. Firstly, getting access to software SMEs to conduct this type of study is very challenging. Therefore, although greater numbers of participating organisations contribute to the generalisability of findings, there exists a practical limitation in getting access to organisations.

A second sample size consideration relates to the time and resource demands required to conduct any given study. In the case of this research, and as summarised in Figure 6, three separate phenomena are under examination in each of the participating organisations. Investigating these three phenomena may require gaining access to three different sets of personnel in the participating companies. In addition to compounding the challenge of securing participating companies, three separate investigations represents a very large burden for the interviewer (not just in terms of basic interviewing time, but also in terms of logistics, travel, organisation, cost and communication). Therefore, there is also a practical limitation on the number of interviews that one researcher can reasonably expect to conduct.

Taking the noted sample size considerations into account, this research initially pursued the strategy of convincing as many organisations as possible to participate in the study. The challenge of getting access to companies was very evident at this stage in the company identification process. Since many software SMEs are intensely busy, the strongest source of candidate organisations resided in the personal contact network of the researcher. Without such a network, it is highly unlikely that a study of this type could be conducted – since a researcher needs to have previously established a respectful relationship with a participant in order for them to commit to an investigation of this type. Therefore, using the personal contact network of the researcher (and extensions to the network formed through existing contacts), seventeen separate software SMEs initially agreed to participate in the study.

3.7 Summary

This chapter presented the principal research philosophies and their corresponding methods and methodologies, establishing that the use of interviews that draw on methods from differing philosophical dispositions represents the best approach to examining the research hypotheses and resolving the research questions posited by this study. Finally, the issue of determining cause-and-effect was discussed.

Comprehensive details regarding the approach to data analysis are presented later in this thesis, in chapter 8.

In the following section, Part 3, the data components are examined in greater detail and the data collection mechanisms are outlined.

Part 3 Data Components and Data Collection

The third part of the thesis contains four chapters. Chapters 4, 5 and 6 provide details of the reference frameworks employed when examining the three principle data components in the study: amount of SPI activity, degree of situational change, extent of business success. These three chapters also present details on how the chosen reference frameworks are systematically transformed into survey instruments that are suited to the task of examining each of the data components. The final chapter of Part 3 of this thesis, chapter 7, presents details regarding the data collection, including the time periods during which the data was collected, the participating organisations and the study participants.

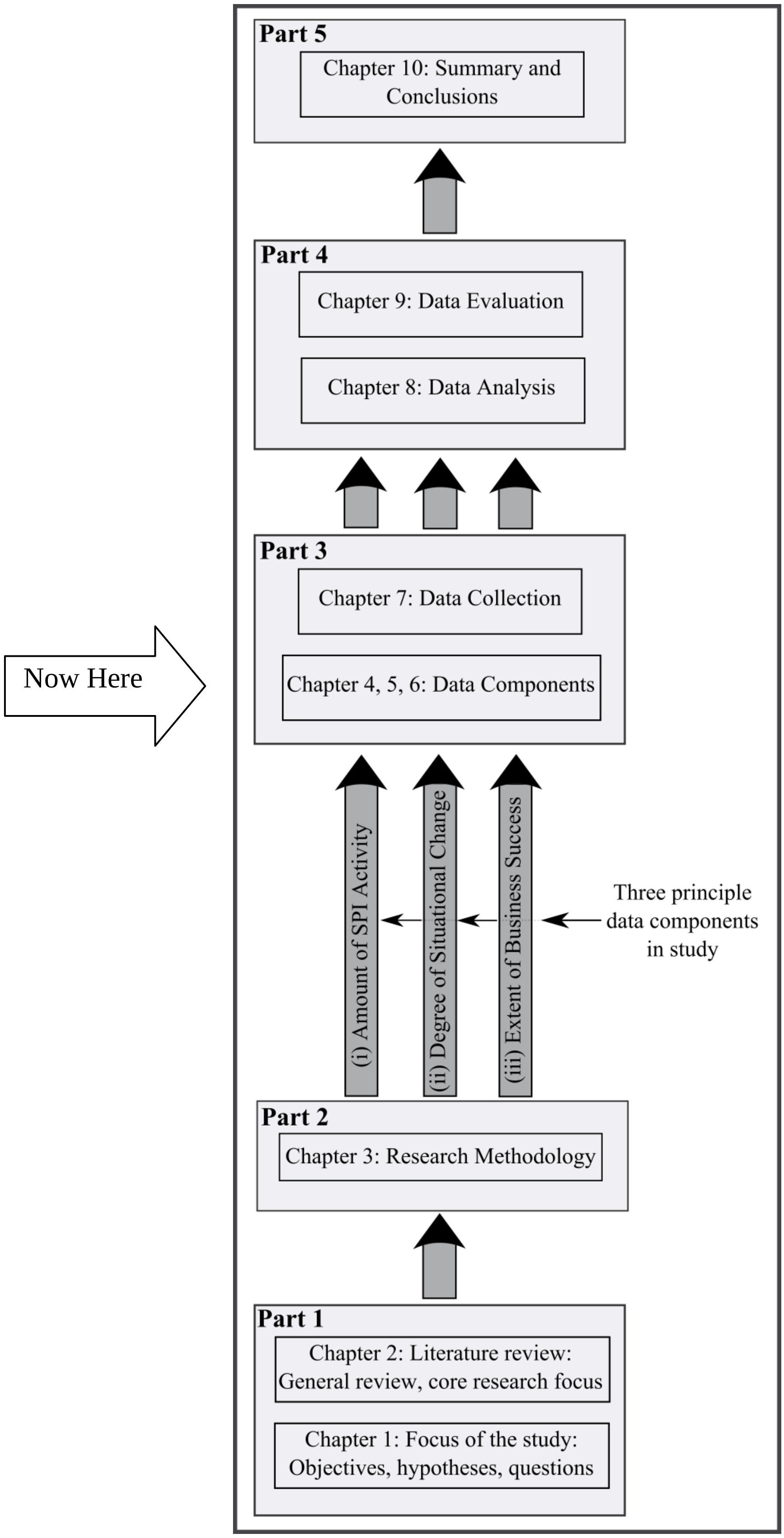


Figure 7 Map of Thesis – Part 3

4 Examining the Amount of SPI Activity

4.1 Introduction

This chapter outlines the general approach to examining the amount of SPI activity in the participating organisations, followed by details of the process for selecting a reference framework for undertaking the SPI examination. Thereafter, the technique for transforming the chosen reference framework into a survey instrument is outlined, along with details of the transformation technique application.

4.2 Process maturity versus SPI activity

As outlined in chapter 2, the process maturity concept relates to the extent to which a software development process is mature as compared with a maturity range set out at either an individual process level or at an organisation-wide level. While the process maturity concept is related to the amount of SPI activity, it is also different in one key respect: SPI activity is concerned only with the amount of SPI implemented in an organisation over a period of time and is therefore divorced from process maturity considerations. While at an abstract level software development organisations have a recurring requirement to improve their software development process (if for no other reason than to adapt to their changing circumstances), it seems likely that no one software development or SPI approach, including process maturity, is universally useful or universally implemented. Ideally, therefore, any study of SPI activity alone should not limit itself to just one software maturity framework or methodology. Nonetheless, it is worth considering if the process assessment vehicles that are associated with software process maturity frameworks might be of some assistance in conducting the SPI activity investigation.

The amount of SPI activity in an organisation could be determined – if only indirectly – through the use of two separate process maturity assessments. A first process maturity assessment could be utilised in order to develop a process maturity profile at a point in time. After some time had elapsed, a second process maturity assessment could be conducted so as to determine the process maturity profile at a second point in time. By conducting a differential analysis of the two process maturity profiles, it would be possible to determine the amount of SPI activity implemented in the period between the two process assessment dates. Although the amount of SPI activity could be determined in this manner, there are a number of strong motivations for not adopting this approach in this research. First and foremost, the use of two process maturity assessments represents a convoluted method for determining the amount of SPI activity. The primary objective of a process maturity assessment is not to collect data regarding the amount of SPI activity, but rather to identify the process maturity level or profile. Therefore, process maturity assessments represent an inefficient approach to determining the amount of SPI activity in an organisation.

A second motivation for not using process maturity assessments to examine SPI activity relates to the sector under investigation in this research: software SMEs. Process maturity frameworks are not commonly adopted in software SMEs, with earlier studies demonstrating that the SME sector considers such frameworks as being infeasible for implementation in their organisations (Staples et al. 2007). Therefore, if process maturity assessments were to be utilised in this research, there is a risk that participating software SMEs would view the research as unsuited to their organisation and opt out of the study. The time requirement for conducting process maturity assessments also contributes to their unsuitability for this study. As outlined in chapter 3, there is a large burden of effort associated with the data collection requirements of this research – with three separate phenomena under investigation in multiple organisations. This burden is shared by both the researcher and the participants, and it very much undesirable to extend this already large data collection burden to include data that is not directly relevant to the study. The research is therefore strongly motivated to be as efficient as possible in making reliable determinations in relation to all of the phenomena of interest.

While two process maturity assessments could be adopted in this study, they would represent a decidedly inefficient approach to examining SPI activity alone. Furthermore, software SMEs have tended not to implement process maturity frameworks, guided by the view that they are infeasible (i.e. overbearing) for their needs. Therefore, the challenge of securing participating organisations would most likely be increased were process maturity assessments to be proposed to candidate participants. Consequently, this research determined to examine alternative, more efficient and more suitable approaches to examining the amount of SPI activity in software SMEs.

4.3 Examining SPI activity

Considering the limitations of using two process assessments to determine the amount of SPI implemented in an organisation, this study developed a new approach to examining SPI activity that requires just a single engagement with the participating organisations. Since this study is concerned with examining the broadest possible spectrum of SPI activity, the study must as a pre-requisite employ a robust and reliable software development process reference framework.

When selecting the software development process reference framework for this study, a number of considerations had to be satisfied. Firstly, the chosen process reference framework had to be comprehensive in nature, identifying and describing the broadest possible range of software development related processes. Secondly, since a number of different software SMEs will participate in the study, it is important that the selected process reference framework should be independent of any specific software development approach – it should identify the process components rather than prescribe the process implementation. Thirdly, in order to maximise the credibility of the research, insofar as is possible the chosen process reference framework should be consensually agreed and generally accepted in the software development community. While a number of possible reference frameworks could be harnessed in order to conduct this study, no single framework addresses the three considerations more completely than ISO/IEC 12207.

According to ISO/IEC 12207, the steps involved in the lifecycle of software development belong to one of two classifications: *software specific* processes and *system context* processes. Software specific processes concern the activities directly related to the core software development effort, such as constructing detailed designs and writing code. System context processes relate to all the non software specific activities that are needed in the broader lifecycle of systems development, such as project planning and systems operation. ISO/IEC 12207 identifies 43 software specific and system context processes, with over 400 corresponding tasks, and is therefore considered to be comprehensive in nature. Thus, ISO/IEC 12207 satisfies the first consideration for the software process reference framework for this study.

ISO/IEC 12207 "*describes [the] continuing responsibilities that must be achieved and maintained during the life of the process… the functions to be performed rather than organizations to execute them*" (Moore 1998 p328), and as such it provides a "*meta-model that defines common software engineering activities independently of a particular life-cycle model*" (Tilley et al. 2005 p251). Therefore, ISO/IEC 12207 satisfies the second consideration for the software process reference framework for this study. Regarding the third consideration for the process reference framework for this research, the approach of the International Organisation for Standardisation (ISO) to drafting and accepting standards involves a democratic voting system, wherein at least 75% of the participating national bodies must approve a standard prior to publication (ISO/IEC 2008). Furthermore, ISO/IEC 12207 was developed in collaboration with the Institute of Electrical and Electronics Engineers (IEEE) Computer Society. As a result, ISO/IEC 12207 satisfies the third consideration for the software process reference framework for this study.

Given the comprehensive nature of ISO/IEC 12207 in identifying the atomic process components for software development, and the benefit of broad international participation in the development and maintenance of the framework, this research selected ISO/IEC 12207 as the preferred software development process reference framework.

4.4 Survey instrument for examining SPI activity

The creation of a survey instrument based on ISO/IEC 12207 needs to be structured and systematic. This section presents the approach adopted when converting an international standard into a survey instrument, followed by an explanation of how the method was applied in the case of transforming ISO/IEC 12207 into an appropriate survey instrument for evaluating the amount of SPI activity in an organisation.

4.4.1 Method for converting an international standard into a survey instrument

Many international standards consist of verbose text that seeks to accurately and completely describe an item of technical matter. However, such comprehensive text-based descriptions are not easily fashioned into survey instruments, especially when practical considerations, such as the time required to conduct a survey, are taken into consideration. Therefore, this research developed a technique for resolving verbose text-based international standards back to comprehensive, yet practical, survey instruments (Clarke and O'Connor 2010). An overview of this technique is presented in Figure 8.

The initial phase, the *Review and Tag* phase, involves reviewing the international standard document, so as to develop a thorough understanding of all the material comprising the standard. Thereafter, the various components of the international standard are tagged – in order to identify the key activities. This requires that close attention is paid to all actions in the international standard, ensuring that no important detail is overlooked.

Following the tagging exercise, the *Question Development* phase is undertaken. This is a four-step activity that involves transforming the tagged details, as output from the initial phase, into a representative, accurate, comprehensive and readable survey instrument. Notes that explain any modifications, along with rationale for changes, must be maintained at each step in the question development phase – this allows for later examination of the survey construction exercise, including the possibility of auditing the artifacts so as to verify that appropriate decisions have been taken throughout the survey construction activity. Such artifacts can thereafter be published along with the survey findings if required.

The first step of the *Question Development* phase involves using the tagged details in order to derive a baseline set of questions. This results in a baseline suite of questions that preserve all of the essential details that are present in the international standard itself. In the second step of the *Question Development* phase, the baseline suite of questions is desk-checked so that any duplications or areas of overlap are resolved. This is necessary in order to efface cross-references that can exist in international standards.

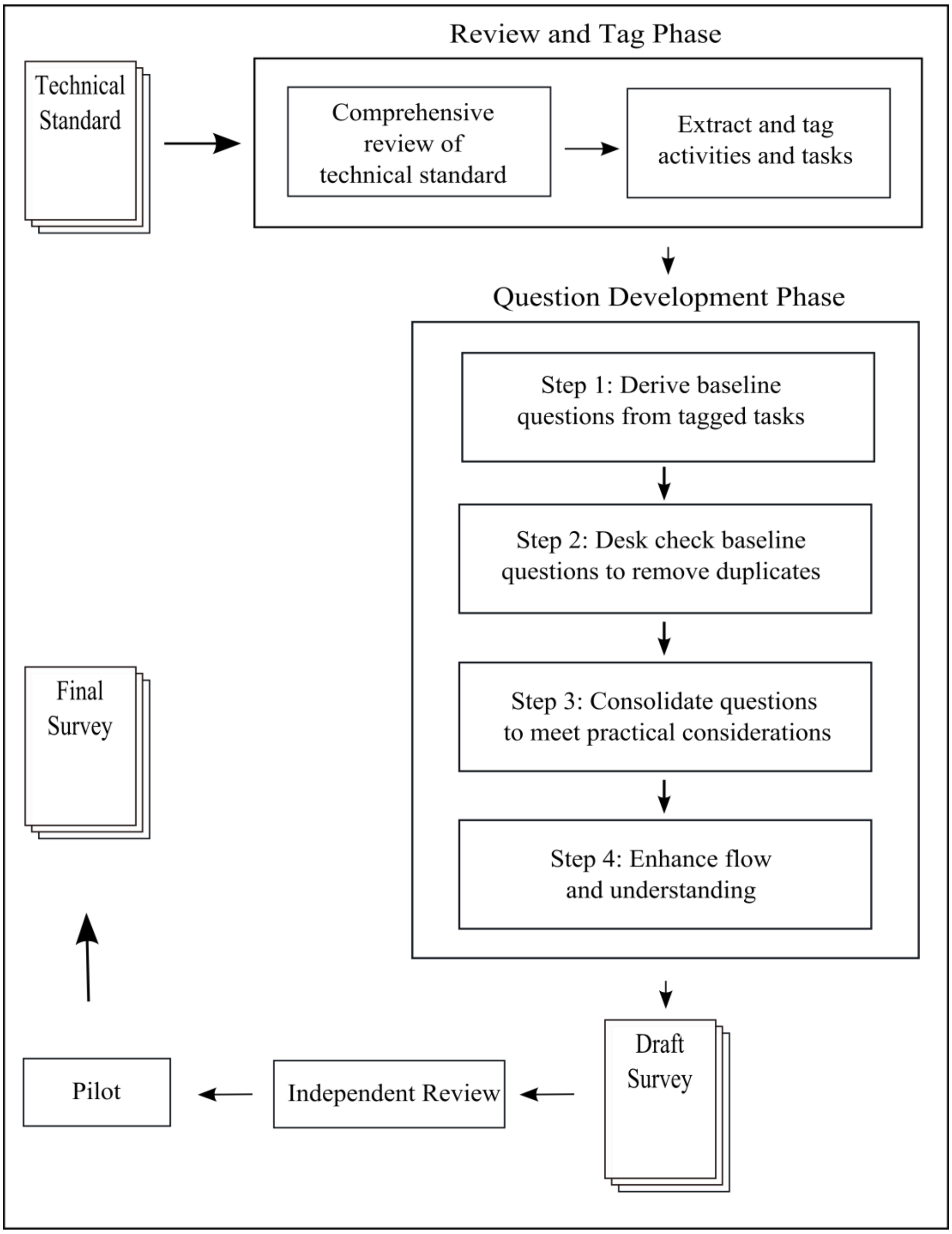


Figure 8 SPI activity survey instrument development technique

The third step of the *Question Development* phase consolidates the list of questions with respect to practical considerations. The target survey duration is among the practical considerations, and the survey constructor must give careful consideration to the appropriate type and number of questions for the survey. The consolidation of questions also requires a considerable deal of knowledge, coupled with expertise, on the part of the survey constructor, but should nonetheless seek to preserve the original makeup and structure of the international standard, retaining all major components such that the resulting survey is clearly identifiable as a derivative of the original standard. Having consolidated the questions in an appropriate fashion, the fourth and final step of the *Question Development* phase involves reviewing the survey so as to enhance the clarity of individual questions and to optimise the flow of the survey so as to best achieve the survey objectives.

Having completed the *Question Development* phase, the survey constructor presents a draft version of the survey instrument to domain experts so as to elicit independent feedback on the content, accuracy, and likely effectiveness of the interview in obtaining the required information. During this *Independent Review* phase, it is important that recognised and credible domain experts are identified and engaged with the review process. Ideally, such independent experts should be drawn from the broader international community, should have unparalleled expertise concerning the underlying international standard, and should have extensive personal experience in the software development domain. Clearly, getting access to domain experts with this type of profile can be hugely challenging. However, if the survey instrument is to be successful in addressing its fundamental purpose, and if it is to withstand the critical review of academic peers and practitioners, it is essential that it is subjected to a robust independent expert review, with selected reviewers meeting the highest possible levels of expertise in relation to the international standard.

The reviewers should be directly queried on all essential aspects of the survey instrument, including the suitability of the instrument in terms of fulfilling its purpose. Reviewers should also be asked to directly comment on the extent to which the survey instrument incorporates the underlying standard components. Furthermore, it is important that this engagement of expert reviewers is formal in nature, with a structured and effective reviewer feedback mechanism put in place.

Following completion of the *independent expert review*, the survey instrument should be revised so as incorporate the feedback from the expert reviewers. Once again, a copy of the changes applied should be maintained so as to allow for later examination of the technique. The final step in the survey development technique involves the use of a *Pilot* deployment with an industrial partner, so as to elicit feedback on the flow, executability and efficacy of the survey instrument. Again, this feedback is integrated into an improved version of the survey instrument.

4.4.2 Application of conversion method to ISO/IEC 12207

This section outlines the application of the survey instrument development technique, presented above, to the development of an SPI activity survey instrument based on the ISO/IEC 12207 international standard.

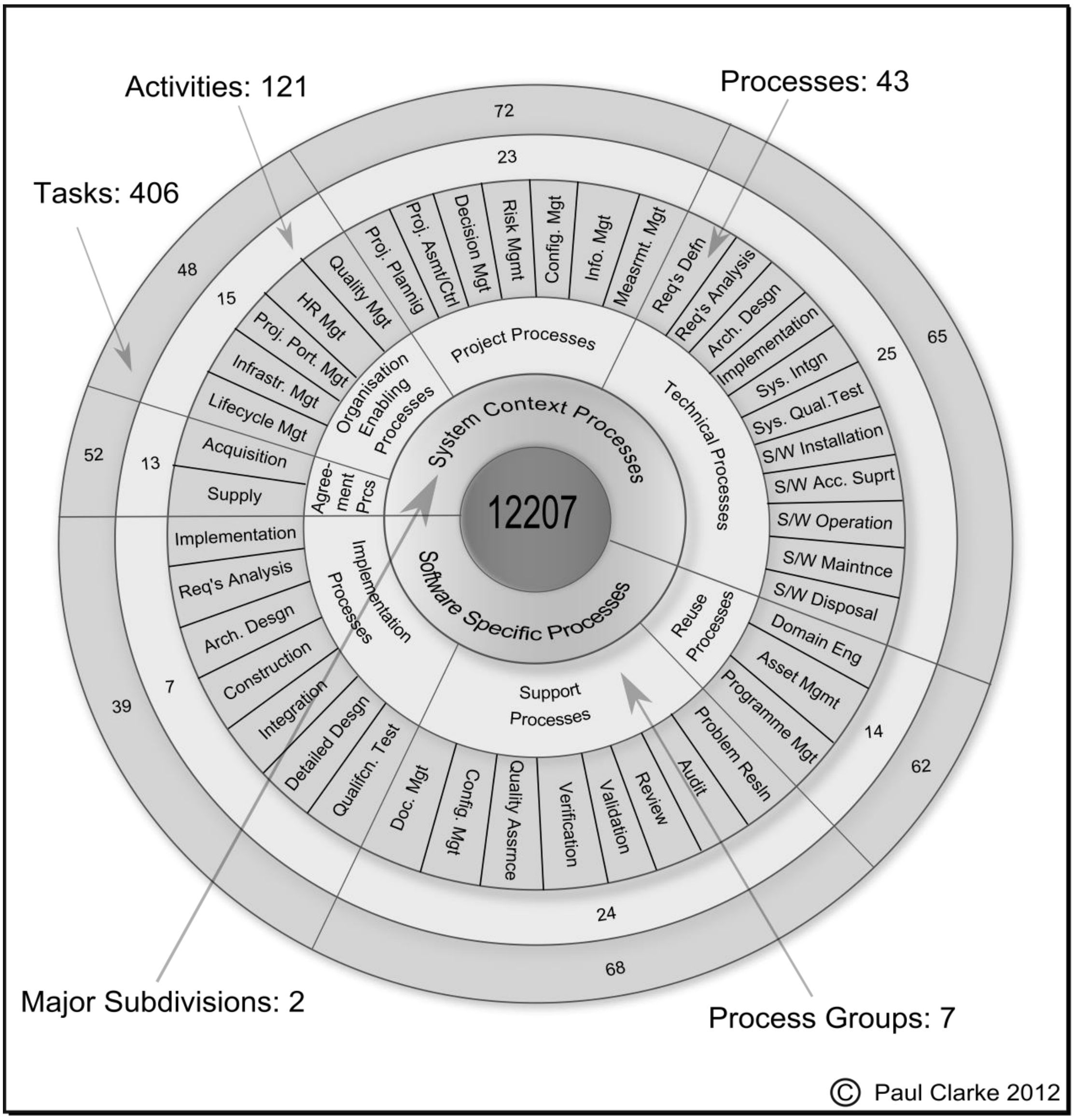


Figure 9 ISO/IEC 12207 topology

4.4.2.1 Review and tag phase

A comprehensive review of ISO/IEC 12207 reveals that the standard consists of seven process groups, forty three processes, one hundred and twenty one activities, and four hundred and six individual tasks. Tasks represent the finest level of detail, with ISO/IEC 12207 defining a task as a *requirement, recommendation, or permissible action, intended to contribute to the achievement of one or more outcomes of a process*; while an activity represents a grouping of *the set of cohesive tasks of a proces*s. The topology of ISO/IEC 12207 is outlined in Figure 9.

The completion of the comprehensive review of ISO/IEC 12207 and the development of a clear understanding of its constituent parts permitted the commencement of the tagging stage. For the purpose of illustration, one of the forty three processes, the *Software Implementation* process, is used to demonstrate the development of the survey instrument from its original form as a list of activities and tasks in ISO/IEC 12207 into its final rendering as a set of questions in a survey. In ISO/IEC 12207, the *Software Implementation* process consists of one activity, the *Software Implementation Strategy*, which is further broken down into five individual tasks. These tasks are tagged, as shown in the highlighted text in Figure 10.

4.4.2.2 Question development phase

Step one of the question development phase involves the construction of a set of baseline questions using the tagged task items from the initial review and tag phase. The researcher, being a person with extensive industrial experience in software development, was able to apply expertise and domain knowledge in developing questions that incorporated the core and fundamental aspects of each software development process. The result was a set of 173 baseline questions that accurately captured the process descriptions in ISO/IEC 12207. In the case of the *Software Implementation* process, the baseline set of questions are as depicted in Figure 11.

As per the survey instrument development technique described earlier, the baseline questions are desk checked to remove any duplicate items. In the case of the software implementation process baseline questions, the life cycle model, software documentation, configuration management, problem resolution, change control, support processes and the establishment of baselines items are all covered in more detail elsewhere in ISO/IEC 12207. For example, the configuration management, problem resolution, documentation management, life cycle management, and support processes are all afforded their own explicit process in ISO/IEC 12207. Consequently, these items are identified as duplicates and removed from the software implementation questions. This recursive and detailed examination of ISO/IEC 12207 and the associated question development required a very considerable amount of time and effort. As indicated earlier, ISO/IEC 12207 has in excess of 400 interrelated tasks, and therefore the development of a concise, yet comprehensive set of baseline questions was no minor undertaking. At the end of step 2 of the question development, there was concentrated set of 143 questions. The resultant question for the *software implementation* process is as depicted in Figure 12.

			Tasks
Software Implementation	**Activity:** Software Implementation Strategy	1	If not stipulated in the contract, the developer shall **define** or select a **life cycle model** appropriate to the scope, magnitude, and complexity of the project
		2	The implementer shall: a) **Document** the **outputs** in accordance with the Software Documentation Management Process; b) Place the **outputs** under the **Software Configuration Management** Process and perform **change control** in accordance with it; c) **Document** and **resolve problems** and **non-conformances** found in the software products and tasks in accordance with the Software Problem Resolution Process; d) **Perform supporting processes** as specified in the contract; e) **Establish baselines** and **incorporate configuration** items at **appropriate times,** as determined by the acquirer and the supplier
		3	The implementer shall **select, tailor,** and **use** those **standards, methods, tools,** and computer **programming languages** (if not stipulated in the contract) that are documented, appropriate, and established by the organisation for performing the activities of the Software Implementation Process and supporting processes
		4	The implementer shall **develop plans** for conducting the activities of the **Software Implementation** process. The plans should include specific **standards, methods, tools, actions,** and **responsibility** associated with the **development** and **qualification** of **all requirements** including safety and security
		5	**Non-deliverable items** may be employed in the **development or maintenance** of the software product

Figure 10 Software implementation process tagging

Regarding Software Implementation, describe any modifications that have been applied to the approach to:

- The life cycle model definition
- Documenting the software
- Configuration management of outputs
- Performance of change control
- Documenting and resolving problems and non-conformances discovered in the software
- Performing supporting processes as defined in contracts
- Establishing baselines and incorporating configuration items at appropriate times
- Selecting, tailoring and using standards, methods, tools and programming languages
- Developing plans for software implementation, including standards, methods, tools, actions and responsibilities associated with the development and qualification of all requirements
- The employment of non-deliverable items in the development or maintenance of the software product

Figure 11 Software implementation process: question development step 1

Regarding Software Implementation, describe any modifications that have been applied to the approach to:

- Selecting, tailoring and using standards, methods, tools and programming languages
- Developing plans for software implementation, including standards, methods, tools, actions and responsibilities associated with the development and qualification of all requirements
- The employment of non-deliverable items in the development or maintenance of the software product

Figure 12 Software implementation process: question development step 2

Step 3 of the question development phase involves the reduction of the question burden in order to meet practical considerations. Given that a number of other data collections were required in the broader study (i.e. situational change and business success data) the SPI activity survey instrument was further developed in order to require approximately two hours to discharge. As a result, and on the assumption that a single question will take on average 1.5 minutes to complete, eighty-five individual questions was identified as the threshold and target for the question burden in the final SPI activity survey instrument.

The numbers presented here are worthy of some further justification. In the case of the time requirement for discharge of the SPI activity survey instrument, it was necessary to be cognisant of the overall time requirement for the study data collection. If the data collection time requirement was too high, it would not be possible to secure the participation of organisations. Equally, if the overall time requirement was too large, it may not be practically possible for a single researcher to conduct all of the data collection. Bearing this consideration in mind, it was the view of the researcher the overall time requirement per participating organisation should not exceed one regular working day. Despite the fact that the time requirement imposed an artificial practicality on the survey instrument duration, a significant amount of focused data was collected. For example, using eighty-five questions, each and every one of the forty three processes in ISO/IEC 12207 can be addressed, and with some additional spare room for manoeuvre. As a guiding principle, it was resolved that the original forty three processes of ISO/IEC 12207 should be preserved and distinctly identifiable in the final survey instrument.

Regarding Software Disposal, describe any modifications that have been applied to the approach to:

- Defining and documenting a software disposal strategy
- Executing a software disposal plan
- Notifying users of the plans and activities for the retirement of software products and services
- Operating retiring and new software products in parallel for smooth transition to a new system
- Notifying all concerned parties regarding the scheduled retirement time
- Accessing data associated with retired software products in accordance with contract and data protection/audit requirements

Figure 13 Software disposal process: question development step 2

During the *Software Implementation* process question reduction, no reductions were performed on the questions that were output from step two of the question development phase. However, question reduction was carried out in other areas of the survey instrument. For example, six *Software Disposal* process baseline questions were resolved back to a single question – as depicted in Figure 13 and Figure 14. The rationale for this reduction is based on the experience and knowledge of the researcher, who has previously observed that in practice, software disposal is not likely to be a detailed and organised event for software SMEs. However, the single question that is retained ensures that the software disposal process is not overlooked in the survey, and the retention of many of the keywords from the original six questions provides for suitable trigger points for survey participants. In this way, much of the concentration of detail from ISO/IEC 12207 in relation to software disposal is retained while also satisfying the practical survey duration consideration.

Regarding Software Disposal, describe any modifications that have been applied to the approach to:

- Defining and executing a software disposal strategy, which may include the parallel operation of retiring and new systems, the notification of associated activities, and the control of access to data associated with retired software products in accordance with contract and data protection/audit requirements

Figure 14 Software disposal process: question development step 3

	Number of Questions		
Grouping	**Step 1**	**Step 2**	**Step 3**
Agreement Process	6	4	4
Organisational Project-Enabling Processes	15	14	10
Project Processes	23	21	13
Technical Processes	27	21	12
Software Implementation Processes	60	53	30
Software Support Processes	28	20	14
Software Reuse Processes	14	10	6
Total	**173**	**143**	**85**

Table 2 Question development and consolidation

The gradual development of the survey instrument up to the completion of step three of the question development phase has witnessed a gradual and careful consolidation of the detailed task information in ISO/IEC 12207 into a survey instrument that can be practically discharged. The various versions of the survey instrument development are preserved in the event that later verification is requested and Table 2 provides a summary of the question development activity.

The fourth and final step of the question development phase involves the adjustment of the survey instrument in order to improve the understandability and flow of the survey. This involved the review of each individual question and where appropriate the reordering or redrafting of individual questions – without removing any detail – so as to render the question easier to understand. Regarding the *Software Implementation* process presented in Figure 12, the questions were reviewed and updated with a view to ease of understanding, the results of which are presented in Figure 15.

Regarding Software Implementation, describe any modifications that have been applied to the approach to:

- Selecting, tailoring and using standards and methods
- Planning for software implementation, identifying the actions and responsibilities associated with the development and qualification of all requirements
- The employment of non-deliverable items in the development or maintenance of the software product, for example programming languages and tools such as software building tools

Figure 15 Software implementation process: question development step 4

In addition to improving the readability and understandability of individual questions, the survey instrument was re-shaped so as to sequence the questions in a manner that addressed specific details towards the start of the survey, with more general questions placed later. For example, the *Software Implementation* process, a detailed and specific process is placed at the start of the survey, while broader processes such as *Human Resource Management* and *Infrastructure Management* are placed towards the end of the survey. Ordering the questions in this way permits the elicitation of specific details earlier in the survey while also allowing for broader question discussion later in the survey (at which stage the specific details are better understood).

In order to validate the draft SPI activity survey instrument, it was subjected to review by independent experts in software processes and in ISO/IEC 12207. When selecting desirable expert reviewers in terms ISO/IEC 12207, by far the most qualified individuals were those involved with the ISO/IEC 12207 standard development and evolution, especially the editorial committee. Fortunately, the researcher was able to access and secure the participation of both the current and a former editor of ISO/IEC 12207, along with two additional sitting members of the committee responsible for maintaining ISO/IEC 12207.

Each of the independent expert reviewers was provided with a copy of the draft survey instrument and an accompanying expert review form. As well as outlining the purpose of the survey instrument, the expert review forms indicated what was required from the reviewer. In addition to being experts in ISO/IEC 12207, collectively the reviewers had accumulated in excess of 45 years academic experience and more than 65 years industrial software development experience. The reviewers were asked a series of questions that focused on three particular areas: (1) How well the survey instrument represented the ISO/IEC 12207 standard; (2) How useful the survey instrument was likely to be in making determinations in relation to SPI activity; and (3) How appropriate the survey instrument was for software SMEs.

In relation to the first area, all four of the expert reviewers agreed that the survey instrument preserved the structure and components of ISO/IEC 12207. Concerning the second and third areas, the expert reviewers made an important recommendation concerning the application of the survey instrument in examining the amount SPI activity in software SMEs. Essentially, it was the view of the reviewers that the *system context* and *software specific* delineation that exists in ISO/IEC 12207 would probably not be applicable to software SMEs. Based on this feedback, a further review of the draft survey instrument was undertaken so that system and software context delineations were consolidated into a single inquiry. For example, the *systems requirements analysis process* and the *software requirements analysis process* were merged into a single line of inquiry.

Following consolidation of the expert feedback, the survey instrument was rendered in its final form, now suited to the task of examining the amount of SPI activity in a software SME over a period of time. In this final form, the survey instrument contains

63 individual questions that query the full spectrum of software process activities as contained in ISO/IEC 12207.

The survey instrument then entered a further validation step, the *Pilot*, during which the instrument was deployed to a software SME. This time, the main purpose of the exercise was to confirm the time required to discharge the survey instrument, and to get live feedback from an industrial interviewee on the experience of participating in the SPI activity survey. At the *Pilot* stage, it was also important to provide the interviewee with an opportunity to comment on any possible oversights in the survey instrument or on any other aspect that was difficult to understand or follow. A candidate organisation was selected for the *Pilot* and the interviewee was the owner and technical leader of a small software development company. The interviewee also had the benefit of approximately 20 years experience in the software development industry, across a number of different domains and involving many different technologies. The feedback from the *Pilot* engagement was largely very positive. The *Pilot* did not generate any additional new process areas that should be investigated for SPI – nonetheless, all later SPI activity surveys included a closing question regarding coverage.

An important piece of feedback from the *Pilot* concerned the need to stress that the timeframe of inquiry was the previous 12 months. The interviewee commented that it was easy to forget that it was just the past year that was under investigation and recommended that this was reiterated throughout all future surveys. This recommendation was taken on board, with the result that at the outset of the final version of the survey instrument, it was stressed that the inquiry was concerned with SPI activity over the past 12 months only. Furthermore, in all subsequent SPI activity interviews, the researcher continually reiterated that the period of investigation was the previous year. Having integrated this feedback from the pilot, the SPI activity survey instrument was considered to be practically deployable to software SMEs en masse. The final version of the SPI activity survey instrument is presented in Appendix B.

4.5 Summary

This chapter presented details regarding the development of a robust approach to determining the first data component of the study: the amount of SPI activity. This new approach to examining the amount of SPI activity has adopted ISO/IEC 12207 as the underlying software development process reference framework and it can be deployed in a single engagement with a participating organisation. In the following chapter, the second data component of the study is presented: the degree of situational change.

5 Examining the Degree of Situational Change

5.1 Introduction

This chapter outlines the approach to examining the degree of situational change in the participating organisations. No earlier publication outlines a comprehensive reference framework of the situational factors affecting the software development process; however, there are a number of related research domains. These related domains are outlined in this chapter, along with a systematic approach to integrating the collective knowledge and wisdom of these sources into an initial reference framework of the situational factors affecting the software development process. Thereafter, the approach to converting the initial reference framework into a situational change survey instrument is presented.

5.2 Background

The various approaches outlined in chapter 1 (ISO 9000, ISO/IEC 15504, CMMI and agile software development methods) offer guidance to the complex problem of software development - with the protagonists from each camp often locked in a bitter battle, each arguing that their particular approach is best. However, despite the significant reported benefits of these approaches, there is a lack of adoption of published models to support the development of software (McConnell 2002, McAdam and Fulton 2002) and therefore, a major gap between theory and practice. Furthermore, attempts to generalise a single software development process for a multitude of settings have met with considerable challenges and the evidence suggests that no single approach to software development "*is universally deployed or even universally useful*" (Jones 2008 p13). An important question to ask is: why is no single approach universally useful? To answer this question, we must examine the basic requirement of a software development process.

According to (Feiler and Humphrey 1992 p6), the basic requirement of a software development process is that it "*should fit the needs of the project*". The needs of the project are informed by the situational context wherein the project must operate and therefore, the most suitable software development process "*is contingent on the context*" (Benediktsson, Dalcher and Thorbergsson 2006 p97). Software development and process managers must "*evaluate a wide range of contextual factors before deciding on the most appropriate process to adopt for any given project*" (MacCormack and Verganti 2003 p230), a view that is shared by Subramanian (2009) who states that the chosen development approach should "*best fit the conditions, product, talent, and goals of the markets and organisations*". Kautz (1998) also shares this view, stating that process improvement initiatives, especially in small companies, should be "*adjusted to their particular situation and… should not slavishly follow one of the comprehensive approaches*".

When it comes to defining a software development process, it therefore seems likely that the claim that "*one size fits all*" is in fact a myth (Boehm and Turner 2003 p7) and that one of the central reasons accounting for this is the rich variation in situational contexts. Although individual situational circumstances are reported as a key consideration when constructing a software development process, there does not exist a general reference framework of the various dimensions of situational circumstance that affect the software development process. Therefore, this research analytically examines a selection of significant contributions from a broad spectrum of related research areas, and uses the output from this analysis to produce a comprehensive initial reference framework of the situational factors that affect the software development process.

5.3 Related domains

Although no general comprehensive reference framework of the situational factors affecting the software process has previously been published, there have been a number of related works. While individually helpful to some extent, none of these related works offers a comprehensive general reference framework of the situational factors affecting the software development process – nor was this their intended purpose. Therefore, an initial reference framework of the situational factors affecting the software development process was constructed, one which incorporates a range of related domains. To avoid the omission of key considerations, the researcher conducted a comprehensive literature review and identified seven distinct related domains that offer data sources that can contribute to the situational factors reference framework. By extensively analysing a selection of significant contributions from seven distinct but overlapping domains, the risk of overlooking key considerations is mitigated – and additional independent reviewing steps described later in this chapter further insure against unintended omissions. While the labelling and classification scheme developed in this work may be open to debate, the rigorous approach to identifying and synthesising a broad range of significant related works ensures that the reference framework is systematically developed and comprehensive. The seven identified domains are summarised in Table 3, along with an explanation of why these domains were selected for inclusion. Within each of the selected domains identified in Table 3, some of the most influential contributions over recent decades are identified as data sources to be included in the analysis process. Primarily, the targeted contributions have had a significant impact on the related research domain. Furthermore, recent contributions, which don't yet have a strong citation record but which are considered by the researcher to offer comprehensive sources for the initial reference framework, are also included in the analysis process.

From the related domains identified in Table 3, a total of 22 individual works were selected for inclusion in the analysis process to render an initial reference framework of the situational factors affecting the software development process. The selected works represent contributions from sources/authors that are commonly accepted as being influential in software engineering, including the IEEE, the ISO, the SEI, Barry Boehm and Alan Albrecht. Furthermore, they span a period of 40 years of research and have accumulated in excess of 5000 citations. While it is not practically possible to consult and integrate every single possible data source for the initial reference framework of the situational factors affecting the software development process, it is essential that a broad spectrum of sources is incorporated and that the selected contributions represent important and substantial contributions in the individual research domains. Adopting this broad yet selective approach to data source identification enables the construction of an initial reference framework that is reasonably comprehensive, credible, and generally useful. Furthermore it provides a foundation upon which to build a more comprehensive reference framework of the situational factors that affect the software development process in the future.

Domain	Reason For Selection
Software development models and standards	Although software development models and standards do not directly seek to identify the situational factors that affect the software development process, they sometimes describe the situational context in a general sense.
Risk factors for software development	The domain of risk factors for software development is concerned with identifying the dimensions of risk that are important considerations for software development projects. Anything that is considered to be a risk for software development is a concern that the software development process may need to take into consideration.
Software development cost estimation	The domain of software development cost estimation is concerned with projecting the costs of software development projects early in the development cycle. Anything that gives rise to a cost in a software project is potentially a consideration for the software development process.
Software development environmental factors	Some research has examined the software development environment and identified a set of characteristics that describe a software development setting. Such characteristics are related to the situational factors that affect the software development process.
Software process tailoring	The domain of software process tailoring is concerned with taking generalised software development approaches and tailoring them to specific software development settings. Consequently, this domain offers some description of the important considerations when tailoring the software process.
Degree of required software process agility	With the advent of agile software development, some research has been conducted that is aimed at identifying the extent to which the software development process should be agile in any given setting. The degree of required agility is an important characteristic for the software development process.
Software engineering body of knowledge	The software engineering body of knowledge (SWEBOK) provides a consensually-validated knowledge repository for software development, and as such, provides some guidance with respect to the factors of situational context that are important considerations for the software development process.

Table 3 Related research domains

5.4 Selected data sources

A review of the literature reveals that currently, there is no single general reference point for the situational factors that affect the software development process. However, as highlighted above, there are a number of different domains that have strong associations with the situational factors that influence the software development process. In selecting data sources for inclusion in the reference framework development process, the researcher applied a multi-faceted approach. Firstly, sources that were highly cited in their particular domain were considered. However, given that works in the domains under examination may only deal tangentally with situational factors that affect the software development process, papers are also identified for inclusion based on other criteria. Therefore, a second selection criterion involved assessing the number of situational factors that are identified in any given work (with a bias towards those contributions that yielded higher numbers of situational factors). A third selection consideration related to the number of references provided in a given data source, since increased references permit a more thorough examination of the meaning of different factors presented in the data source.

In addition, rather than limiting the time period of works for inclusion, a fourth selection criterion permitted that a data source from the more distant past could be selected for inclusion if it was significant in depth of detail and overall contribution (this was quite an important consideration as significant works in the cost estimation and risk factors for software development domains date from earlier research eras). In the following subsections, the related research domains are discussed, explaining their relevance to the exercise of constructing an initial reference framework of the situational factors that affect the software development process. Furthermore, the data sources for inclusion in the initial reference framework construction exercise are identified.

5.4.1 Software development models and standards

In relation to the first domain, software development models and standards, there have been many contributions over the past thirty years. For the purpose of the framework generation analysis process, two of the more extensive and established models and standards are included as data sources: ISO/IEC 12207 (the process reference model associated with ISO/IEC 15504) and CMMI. These two sources have been selected as they represent the two most widely adopted software process maturity reference frameworks, and although they do not seek to identify the full spectrum of situational factors that affect the software process, they do give a high level indication of the general categories of factors that are believed to be important when defining a software development process.

ISO/IEC 12207 recognises that all software development projects are "*conducted within the context of an organization*" (ISO/IEC 2008 p24) and offers two broad classifications of the factors that affect the software development process: *Organisational process outcomes as factors affecting process* and *Project factors affecting the process definition*. These two broad classifications are further broken down into a set of factors that are included as data sources for the construction of the general reference framework. In common with ISO/IEC 12207, CMMI does not present a detailed listing of the situational factors that affect the software development process. However, CMMI does recognise that "*an organization's processes operate in a business context that must be understood*" (SEI 2006 p243). Furthermore, CMMI states that "*issues related to finance, technology, quality, human resources, and marketing are important process considerations*" (SEI 2006 p243). In addition to these important process considerations, CMMI also identifies a set of *derived requirements* that essentially represent the contextual requirements that may not be explicitly stated in customer requirements, but which are inferred (1) from contextual requirements (e.g. applicable standards, laws, policies, common practices, and management decisions), or (2) from requirements needed to specify a product component. These aspects of situational context described in ISO/IEC 12207 and CMMI are included as data sources for the reference framework data analysis.

5.4.2 Risk factors for software development

The second related domain, risk factors for software development, is concerned with identifying the dimensions of risk that are considered important for software development projects. A risk factor "*represents an uncertain attribute of a project or its contextual environment*" (Benaroch and Appari 2010 p66) that can "*adversely affect a project, unless project managers take appropriate countermeasures*" (Wallace and Keil 2004 p70). Such countermeasures can include software development process changes and therefore risk factors are an important data source for consideration when constructing a general set of situational factors that affect the software process.

The area of software development risk is a large and quite distinct research domain that has grown over a number of decades and it is not the intention of this data analysis to consider each and every contribution to the domain. However, it is important that a comprehensive profile of the risk factors for software development is extracted from the existing literature – and the following paragraphs identify the contributions that this work includes so as to achieve such a comprehensive profile.

Casher (1984) presents a project risk framework, identifying 21 risk factors associated with information systems projects. While Casher (1984) identifies a substantial listing of the risk factors associated with software development, there is no attempt to rank the individual risk factors in terms of their priority or to identify what might be considered to be the more important risk factors. This prioritisation of risk factors is an area that Barry Boehm has examined, and his research culminated in the publication of the *Top 10 Software Risk Items* (Boehm 1991). Based on a survey of "*several experienced project managers*", the "*top ten primary sources of risk on software projects*" are identified (Boehm 1991 p429).

Later research into the prioritisation of risk factors finds that Boehm's top ten listing of software risk items (along with much of the pre-existing risk-related literature) is focused on what are termed *execution* risks (Keil et al. 1998). *Execution* risks are considered to be those risk factors over which the project manager has some control. However, Keil et al. (1998) attempts to include a broader spectrum of risks than just the *execution* risks and consequently, the top eleven risk factors identified by Keil et al. (1998) contains a number of risk factors that are absent from the earlier classification compiled by Boehm (1991).

In an attempt to synthesise the best aspects of all pre-existing risk management techniques, Lyytinen et al. (1998) combines the benefits and wisdom of established software development risk management approaches into a single risk management model, a so called *socio-technical* model of risk management, which identifies 17 distinct risk items which are classified under five risk classifications. Ropponen and

Lyytinen (2000) subsequently carry out a survey of 83 project managers in order to determine the components of software development risk and to what extent these components are influenced by environmental factors. This later study by Ropponen and Lyytinen identifies six software risk components (broken down into 26 associated risk items) and four environmental factors that are acting on the risk components. Ropponen and Lyytinen (2000) acknowledge that some of the environmental factors may be beyond the control of the project manager (e.g. setting project deadlines and making decisions relating to subcontracting). Nonetheless, they conclude that even these issues are "*crucial issues to pay attention to and to take actions to reduce risks related to them*" (Ropponen and Lyytinen 2000 p107).

Building upon the work of Ropponen and Lyytinen (2000) and others, Barki et al. (2001) develop a framework of software development risks that encompasses a broader spectrum of risks than the earlier risk frameworks – including new risk items such as *interpersonal conflicts* and *user attitudes*. The framework includes 35 risk factors that are classified under 18 *underlying concepts*. Furthermore, some of the risk factors are further decomposed into a series of components. The broad risk factor framework introduced in Barki et al. (2001) heralded the arrival of more extensive risk factor models. In 2004, a larger framework again (Wallace and Keil 2004) is introduced, this time with 53 risk factors (with considerable overlap with the risk factors identified earlier in Barki et al. (2001)).

More recent work that seeks to develop a method for pricing the risk factors for software development has also identified a list of "*several software development risk factors*" (Appari and Benaroch 2010 p2099). These risk factors are set out in two publications, both from 2010 (Benaroch and Appari 2010, Appari and Benaroch 2010). While these two works provide some recent research on risk factors for software development, they do not offer new additional risk factors over the earlier risk factor related studies.

The software development risk contributions outlined above span a period of almost thirty years and the significant depth of information provided in these contributions is included in the reference framework data analysis.

5.4.3 Software development cost estimation

The third related domain, software development cost estimation, is concerned with projecting the cost of software projects early in the development lifecycle. A number of approaches exist to support the early estimation of the cost associated with a software development effort, with Delany and Cunningham (2008) suggesting that these techniques broadly fall under three principal categories: 1) expert judgement involving predictions based on the skill and experience of one or more experts; 2) analogy involving the comparison of one or more completed projects with details of a new project to predict cost and duration; and 3) algorithmic models which predict estimates of effort and duration using parametric equations.

Expert judgement is considered to be "*consulting with one or more experts, who use their experience and understanding of the proposed project to arrive at an estimate of its cost*" (Boehm 1981 p333), while estimation by *analogy* "*involves reasoning by analogy with one or more completed projects to relate their actual costs to an estimate of the cost of a similar new project*" (Boehm 1981 p336). *Algorithmic* models "*use mathematical formulae to predict effort and duration as a function of a number of variables*" (Delany and Cunningham 2008 p2). Unlike *expert judgement* and *estimation by analogy*, which attempt to leverage the accumulated knowledge and information in individual settings, *algorithmic* models attempt to provide a generic model from which software development costs can be estimated in any setting. As such, *algorithmic* models capture some of the situational factors that can affect the software development cost in a general sense. These cost factors are contained within the *algorithmic* models and are potentially a valuable source of knowledge when constructing a comprehensive reference framework of the situational factors affecting the software development process – since any factor that affects the cost of software development can have a direct influence on the software development process.

According to Kitchenham (1991), there are two categories of *algorithmic* cost estimation models: empirical factor models which provide an estimate of the value of a cost parameter, and constraint models which demonstrate the relationship between the various cost factors. Of the empirical factor models, the Constructive Cost Model (COCOMO) (Boehm 1981) and its later revision COCOMO II (Boehm et al. 2000) are the most widely accepted and used (Heejun and Seung 2008). COCOMO II has 17 cost factors that are considered to represent a conclusive listing of the factors that can affect the cost of a software development effort. Furthermore, COCOMO II includes five additional scale factors that are incorporated into cost estimation: *Precedentedness*, *Flexibility, Design/Risk, Team Cohesion,* and *Process Maturity*. To some extent, these scale factors embody the potential risk and they are applied to the cost as a multiplier of the basic cost convention captured in the accumulated cost factors. Of the constraint models, the Software Lifecycle Methodology (SLIM) (Putnam 1978) is most commonly in use (Delany and Cunningham 2008). The SLIM model is expressed as two equations describing the relationship between the development effort and the schedule. Four basic SLIM parameters and four productivity factors are the essential components of the two equations.

Both the empirical factor models and the constraint models use estimates of the number of source lines of code as a size driver. However, it can be difficult to estimate the lines of code in advance of implementation and an alternative approach, which measures the size of the functionality of the software based on functional specifications, has been proposed as an alternative to empirical factor models and constraint models. *Function Point Analysis* (FPA) (Albrecht 1979) is the first and among the most commonly used functional size measurement approach (Koh, Selamat and Ghani 2008).

The FPA method is based on the idea of determining size based on functional requirements from the end user's viewpoint, taking into account only those elements in the application layer that are logically visible to the user and not the technology used (Albrecht and Gaffney 1983). Application elements, which include things such as external inputs and outputs, are assigned weighting factors which are summed together rendering an unadjusted function point rating. The unadjusted function point rating is scaled up or down based on an evaluation of the application's characteristics.

While leading algorithmic cost estimation models such as COCOMO II, SLIM and FPA provide a mechanism for estimating the cost of software development efforts, there remains a debate as to the effectiveness of using these approaches (Delany and Cunningham 2008). This debate has given rise to some emerging cost estimation techniques, including the application of case based reasoning (CBR). CBR (Kolodner 1993, Watson and Marir 1994) involves matching the current problem against similar problems that have already been encountered in the past and thereafter, using earlier solution strategies to inform engineering solutions to present problems. CBR is therefore a problem solving technique that is based on the reuse of past experiences. Delany & Cunningham (2008) apply this problem solving technique to the software cost estimation problem, claiming that it is a more effective approach to software cost estimation than algorithmic cost estimation models. In order to apply CBR to software cost estimation, it is first necessary to identify the features of the software cost estimation case (or problem). Having conducted a literature review into related material on software cost estimation, Delany & Cunningham (2008) develop a list of case features that are considered to capture the 26 key aspects of the software cost estimation case.

The software development cost estimation contributions outlined above span a period of over thirty years and the significant depth of information provided in these contributions is included in the reference framework data analysis.

5.4.4 Software development environmental factors

The fourth related domain, software development environmental factors, is concerned with identifying a set of characteristics that can describe a software setting. Environmental factors are considered to be the "*factors that characterize a project and its environment*" (Xu and Ramesh 2007 p294) and there have been a number of contributions in this area (Xu and Ramesh 2007, Petersen and Wohlin 2009, Dede and Lioufko 2010, Bekkers et al. 2008). Of these contributions, it is the work of Xu and Ramesh (2007) that is most well grounded in the earlier related publications. Four primary categories of factors are identified in Xu and Ramesh (2007), which encompass the software development environment: *Project*, *Team*, *External Stakeholders*, and *Organisation*. These four categories are comprised of 20 distinct software factors, including such items as *project size*, *project type* and *turnover of personnel*. In addition to identifying these environmental factors, Xu and Ramesh (2007) also present a model of software process tailoring that consists of four distinct dimensions: *Set Project Goals*, *Tailor Process*, *Assess/Adjust Environmental Factors*, and *Evaluate Challenges*. In this process tailoring model, the four dimensions are constructed so as to be interdependent. Changes to the software process can give rise to changes in the environment, and changes in the environment can also give rise to changes in the software process. This bilateral relationship is an interesting concept, as it depicts the software development setting as being quite fluid, where changes in one dimension give rise to changes in another dimension, which may in turn give rise to changes in another dimension again (including changes in the dimension in which the initial change originated). This inter-relationship is considered to be representative of an *amethodical* process (Truex, Baskerville and Travis 2000) rather than a *methodical* process, suggesting that software development processes are highly dynamic, that software development is an opportunistic process that needs to be adapted and tailored constantly during execution of the process.

The software development environmental factors identified by Xu and Ramesh (2007) offer a significant depth of information which is included in the reference framework data analysis.

5.4.5 Software process tailoring

The fifth related domain, software process tailoring, is concerned with taking generalised software development approaches and tailoring them to specific software development settings. As outlined earlier in this chapter, adjustments to standard software processes are necessary to make them suitable for specific environments (Baskerville and Stage 2001). Such adjustments are often referred to as process *tailoring*, an activity which has been described as "*adjusting the definitions and/or particularizing the terms of general description to derive a description applicable to an alternate (less general) environment*" (Ginsberg and Quinn 1995 p3). Process tailoring factors are of relevance when constructing a reference framework of the situational factors affecting the software process, with "*contextual issues… [considered to be among] the main inputs to the tailoring process*" (Coleman and O'Connor 2008 p778).

Cameron (2002) explains that "*the diversity of IT projects frustrates any direct attempt to systematize the processes used for their development. One size just won't fit* *all… All too often, deviation from a standard methodology is seen as an imperfection, as an unwelcome compromise (despite the fact it always happens!)*” (Cameron 2002 p74). Furthermore, Cameron (2002) suggests that “*the many factors that contribute to the variation among projects are exactly the factors that influence tailoring decisions*” (Cameron 2002 p75) and identifies a set of five process tailoring factors. Building on the work of Cameron (2002) and others, Ferratt and Mai (2010) identify a set of six factors affecting process tailoring decisions. Consequently, the software process tailoring contributions of Cameron (2002) and Ferratt and Mai (2010) present another rich data source for inclusion in the reference framework data analysis.

5.4.6 Degree of software process agility

The sixth related domain, the degree of required software process agility, is concerned with identifying the extent to which the software development process should be agile in any given software development setting. Perhaps the most notable contribution in this domain is from Boehm and Turner (2003), who present a model that can be applied in order to determine the extent of agility that is required in a software development setting. The framework identifies five key environmental characteristics that are important when determining the required agility. The basis of the Boehm-Turner model is that by determining the importance of the five key dimensions, it is possible to reach a decision in relation to the degree of agility required in a given project or setting. Included in these dimensions are factors such as *size* and *personnel* – both of which are considered important for Boehm and Turner in assessing the degree of required agility. The degree of agility contribution from Boehm and Turner (2003) therefore offers a further valuable data source for inclusion in the reference framework data analysis.

5.4.7 Software engineering body of knowledge

The seventh and final related domain, the Software Engineering Body of Knowledge (SWEBOK) (IEEE 2004), was created with the purpose of providing “*a consensually-validated characterization of the bounds of the software engineering discipline and to provide a topical access to the Body of Knowledge supporting that discipline*” (IEEE 2004 pxix). It is designed to “*serve as a compendium and guide to the body of knowledge that has been developing and evolving over the past four decades*” (IEEE

2004 pvii) and recognises that "*the context of the project and organization will determine the type of process definition that is most useful*" (IEEE 2004 p9-3). SWEBOK distinguishes between the factors that are important when considering the efficacy of the software process and the important variables affecting the software process definition, and provides a listing of the important factors for consideration when defining a software development process. As such, SWEBOK provides an additional important data source for inclusion in the reference framework data analysis.

5.4.8 Summary

This section has outlined the research areas and contributions that represent strong sources of data for the development of an initial comprehensive reference framework of the situational factors that influence the software development process. The data sources identified represent some of the most significant contributions, from some of the most significant contributors over the past four decades. Furthermore, recent contributions that are considered to be of direct relevance but have not yet had the time to build up a strong citation record are also included in the data source listing.

In building an initial reference framework of the situational factors affecting the software development process, it is important to identify and integrate the collective wisdom from related research domains. However, given the large volume of contributions over the past forty-plus years, it is not practically feasible to consult every single possible source from every research domain – nor is it necessary (since there is often a strong overlap of data within individual research domains). It is, however, important that a broad set of significant and comprehensive contributions are selected from the related research domains – and this is the philosophy that this research has adopted.

This section has described the seven related research domains that were selected for inclusion in the data analysis associated with the reference framework construction. Applying the selection criteria identified at the start of this section, 22 individual contributions were selected and examined. These 22 contributions have accumulated in excess of 5000 citations and have provided almost 400 individual data items that can now be used to construct an initial reference framework of the situational factors that influence the software process definition.

5.5 Situational factors framework development methodology

Having identified a broad set of comprehensive works for use in constructing an initial general reference framework of the situational factors affecting the software process, next the data was distilled from the various sources into a core set of situational factors that affect the software process. While the data in the various sources is highly relevant to the situational factors framework construction, there is an inconsistent use of language, and there is a wide variety of content and classifications. It is therefore necessary to analyse the data sources and generate core concepts from what are disparate sources. Such an analysis requires that patterns of meaning are systematically developed from the data, with the result that the data is gradually transformed into a coherent categorisation of situational factors.

The fundamental requirement of the framework development methodology therefore is that it can reliably develop the core meaning from the various different sources. In order to achieve such a goal, it will be necessary to thoroughly review all of the data sources, extracting all the key related factors. Following the factor identification and extraction, the next requirement will be to compare and contrast the factors from the various related sources, with a view to identifying common themes and meanings. The process of comparison and meaning development is recursive in nature, and requires supporting structure so that the researcher can gradually and systematically identify the commonality and differences across the various data sources. Such supporting structure is provided for in the *constant comparison* and *memoing* techniques commonly used in Grounded Theory research (Glaser and Strauss 1976).

5.5.1 Using constant comparison to develop the meaning of data

Grounded Theory is a general methodology of analysis linked with data collection that systematically applies a set of methods to generate an inductive theory about a substantive area (Glaser 1992). The purpose of Grounded Theory is to "*build theories from data about the social world such that theories are 'grounded' in people's* *everyday experiences and actions*" (Knigge and Cope 2006 p2024). Grounded Theory "*emphasises the systematic approach to data collection, handling and analysis*" (Douglas 2003 p44). This research seeks to adopt a similarly systematic approach to data handling and analysis and therefore borrows three key data management and analysis techniques from Grounded Theory: (1) coding, (2) constant comparison, and (3) memoing.

In Grounded Theory, data analysis involves what is commonly termed coding, "*taking raw data and raising it to a conceptual level*" (Corbin and Strauss 2008 p66). Coding involves interacting with data using techniques such as asking about the data, making comparisons between the data, and in doing so, deriving concepts to stand for those data, then developing those concepts in terms of their properties and dimensions (Corbin and Strauss 2008). Therefore, essentially coding is "*the process of defining what the data is about*" (Bryant and Charmaz 2007 p605), of "*deriving and developing concepts from data*" (Corbin and Strauss 2008 p65), whereby "*codes capture patterns and themes and cluster them under an evocative title*" (Lempert 2007 p253). For this research, the initial data codes are the individual identifiers and classifications that each of the selected data sources uses to identify the key atoms of data and their associated concepts.

In Grounded Theory, *constant comparison* refers to "*the analytic process of comparing different pieces of data for similarities and differences*" (Corbin and Strauss 2008 p65). This method of analysis "*generates successively more abstract concepts and theories through inductive processes of comparing data with data, data with category, category with category, and category with concept*" (Bryant and Charmaz 2007 p607). This type of comparison is considered essential to all analysis because it allows the researcher to differentiate one category or theme from another and to "*identify properties and dimensions specific to that category or theme*" (Corbin and Strauss 2008 p73). In the case of this research, the various data codes provided by the selected data sources must be compared for similarities and differences through constant comparison until a final rendering of core factors and categories emerges – this constant comparison is in effect a distillation process with successive iterations producing a more perfected integration of the underlying component data parts. Data coding using constant comparison becomes quite complex, since "*simultaneously* *many categories and their properties may be emerging at different levels of conceptualization and [with] different ways of being related by theoretical codes*" (Glaser 1998 p144). Glaser (1998) recommends that this build-up of complexity is kept track of by the *memoing* process.

The coding of data in Grounded Theory occurs "*in conjunction with analysis through a process of conceptual memoing, capturing the theorist's ideation of the emerging theory*" (Bryant and Charmaz 2007 p265). Memos can be considered to be "*written records of analysis*" and during the process of memoing, a certain degree of analysis occurs (Corbin and Strauss 2008 p117). The very act of memoing "*forces the analyst to think about the data and it is in thinking that analysis actually occurs*" (Corbin and Strauss 2008 p118).

Grounded Theory's tandem process of coding and memoing helps to alleviate the pressure of uncertainty by challenging the researcher to stop and capture, in the moment, their conceptual ideas about the codes that they are finding (Bryant and Charmaz 2007 p275-276). As coding and memoing progress, patterns begin to emerge.

5.5.2 Framework development technique

The task of developing common factors and classifications from the disparate data sources identified earlier shares many of the challenges of a Grounded Theory research effort. Therefore, the core Grounded Theory methods of constant comparison and memoing are employed to ensure a rigorous data consolidation technique.

Earlier in this chapter, several distinct areas within the software development domain were explored in order to determine the type of data that they offered up for inclusion in a reference framework of the situational factors that can affect the software development process. These areas include risk factors for software development, software cost estimation, and factors affecting software development process tailoring. This section now describes the technique that is adopted in order to synthesise the data from these different domains, with an overview of the technique provided in Figure 16.

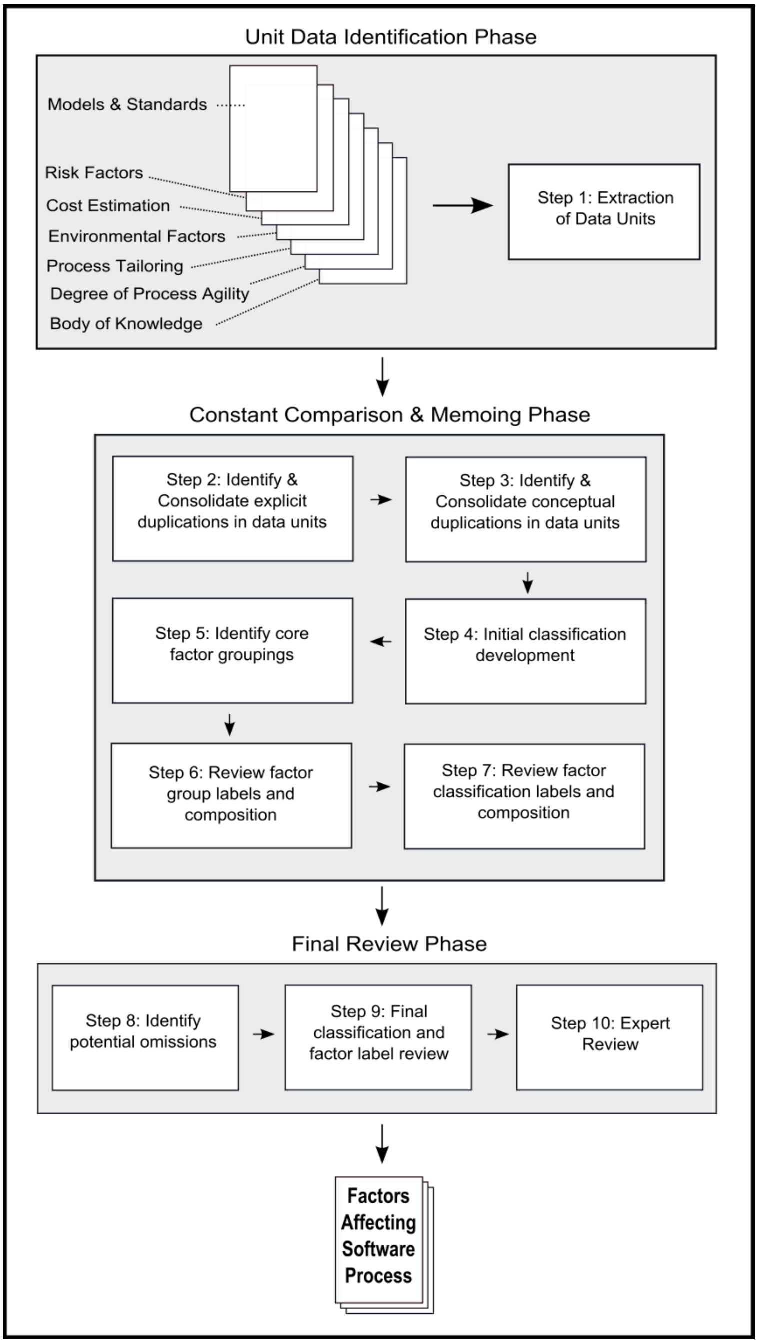


Figure 16 Systematic framework development technique

5.5.3 Phase 1 – Unit data identification

The data sources are examined so as to identify the key details (henceforth referred to as *data units*) that are important considerations when developing a reference framework of the situational factors affecting the software development process. In the case of some of the data sources, this will involve extracting all of the factors in a framework or classification – for example, in the case where there is a listing of cost estimation factors or risk factors. Where frameworks and classifications are not explicitly presented in a data source, this will involve analysing the data source to carefully identify and extract any data that is considered to offer a potential situational factor that affects the software development process. In the first instance, all the data units from all of the data sources are compiled into a master table. This master table represents the data universe upon which the constant comparison and memoing techniques from Grounded Theory will be deployed in order to derive the core factors and classifications. The master list also identifies the original source for each data unit. This originating source linkage is retained throughout the constant comparison process so that the relationship between the eventual factor and classification reference framework and the original sources will be clearly identifiable.

5.5.4 Phase 2 – Constant comparison and memoing

The first phase of the constant comparison seeks to identify those data units for which there are explicit duplications present elsewhere in the data universe. This part of the process involves scanning the data universe to identify instances of duplication. Since there is a one-to-one mapping between the individual data units that are consolidated in this comparison, there is no need to record memos that reflect the thinking that has influenced the consolidation. Where data units are consolidated owing to duplication, care is taken to retain the data source information. Therefore, if two separate data units have the same textual description and meaning, they are merged into a single unit – however, both sources for the data unit will be evident in the consolidated data unit record in the master table.

Following the initial consolidation of duplicated data units, the constant comparison and memoing effort now seeks to identify conceptual duplications or similarities in the data units. It is at this stage that the effort to distil the meaning commences in earnest. The process now permits the consolidation of two data units which do not have the same textual label, but which are considered to embody the same essential meaning. In order to ensure that the understanding of the various sources is sufficient to derive valid consolidated meanings, it is necessary that the researcher develops a thorough prior understanding of the data source. In addition to conducting a robust review of the data sources, the researcher could call upon significant related domain knowledge obtained during a substantial industrial career. This earlier extensive industrial experience of the researcher therefore ensured that a mature and informed degree of knowledge was applied when developing core meanings. Memos are also employed at this stage – both to record and develop the thought process and to ensure that where data items are consolidated, the resulting consolidated data unit retains clear links back to all originating data sources. The resulting consolidated master list of data items can now be considered to be starting to evolve into a systematically consolidated reference framework of the situational factors that affect the software development process.

The next step in the constant comparison involves the construction of an initial classification for the emerging data units. Again, memos record the rationale that has informed the initial classification – so that there is a consistent and traceable vein of thought development present in the master listing. In many cases, the data sources provided a native classification listing for factors, and these classifications are useful for organising and classifying the broader list of factors now contained in the emerging framework. Following the initial classification, the individual data units are consolidated into core factor groupings, each of which is assigned a factor grouping label. These factor groupings are in effect data codes that embody the essential meaning of the individual consolidated data units. As with the earlier stages, memos record the rationale employed in classifying and labelling.

Equipped with a basic form of the factors and associated macro-classifications, the constant comparison effort now revisits the factor groupings and constituent data units in order to evaluate the accuracy of the factor grouping labels and the consistency of the data units that constitute factor groupings. The factor grouping labels are renamed as appropriate and individual data units may be transferred to alternate factors or to entirely new factors – as deemed appropriate with respect to the emerging factors and classification. Memos are once again used as descriptions that help to clarify the thought process and to offer a historical record of the actions taken. As with all previous iterations of the constant comparison, the original sources for data units – many of which are by now consolidated into factor groupings – are retained so that an overview of the master list composition and sources is readily evident. This helps in visualising the sources of factors and classifications, and permits an evaluation of the impact of individual data sources on the overall master list as it emerges.

The classification labels and corresponding factor groupings are now evaluated to determine if the labels are representative of the classification component factors, and also to check that the factors are appropriately classified relative to the emerging master listing. In this iteration, some classifications may be renamed or even consolidated (or broken out) as deemed appropriate. Upon completion of this iteration, the master list is considered to be in a highly emerged state, where the classification and factors are becoming settled in a configuration that is reflective of the emerging themes.

5.5.5 Phase 3 – Final review

The master list is now examined for any possible omissions – items that for some reason have not emerged from the underlying data sources (perhaps due to their absence in the underlying data altogether). While every effort is taken to ensure that a broad and accurate set of data sources is considered in building the set of situational factors that can affect the software development process, it is not possible to guarantee that each and every eventuality is catered for – equally, there remains the possibility that there are situational factors that are important but which have never before been considered for inclusion in the underlying data sources. Again, memos are used to clarify thinking and to act as an historical record of the master list development.

A final factor and classification label review is now undertaken – a step which ensures that each and every item is accurately named. So as to raise the completeness, reliability and utility of the situational factors and classifications, domain experts are asked to conduct reviews. The expert reviewers should have extensive related experience in software development, and should conduct the review with a particular focus on completeness, classification, presentation, and utility. Feedback from these reviews should be applied to the factor and classification framework, raising the overall quality of the reference framework and bringing greater certainty to the product as a whole.

5.6 Application of framework development technique

This section contains details of the application of the technique outlined in the previous section. Much of the memo and constant comparison data associated with the application of the technique is verbose in nature and therefore, this thesis provides just the core details - so as to facilitate an understanding of the systematic and rigorous approach applied in the development of the reference framework. The key output, a reference framework containing the situational factors affecting the software development process is also presented (refer to Table 4, Table 5, Table 6 and Figure 17).

5.6.1 Phase 1 – Unit data identification

During this phase, 397 data units were extracted from the 22 separate data sources identified earlier in this chapter. Each of these units represents a factor that can affect the software development process definition. Furthermore, any classification provided in the data source was also extracted during this phase, such that the baseline raw data listing contains all factors and classifications from all data sources. However, this raw data must be examined for data duplication and analysed for core concept development. In order to support the development of the core concepts and themes, the constant comparison and memoing techniques are adopted.

5.6.2 Phase 2 – Constant comparison and memoing

The six consecutive steps involved in this phase are designed to support the development of common factors and associated factor classifications. The 397 factors produced in the data unit identification phase were gradually and systematically distilled to a set of 48 factors and 11 associated classifications (throughout this distillation process, extensive notes are retained in the form of memos). The naming and classification of factors was heavily influenced by the classifications and terminology that was present in the underlying data sources. By incorporating the factors and classifications directly from each data source, the framework development was guided in a direction that is consistent with commonly accepted terminology and classifications. For example, ten of the data sources contained native classifications for *personnel* or *team* considerations, while seven of the data sources had native classifications relating to *application* considerations. Therefore, as the constant comparison and memoing progressed, both of these areas emerged as strong clusters for distinct classification. Consequently, two of the factor classifications in the initial reference framework are dedicated to these areas: *Personnel* and *Application*.

Other classifications arose from the recurring existence of individual factors in the data sources. For example, although just two of the data sources had native classifications related to *requirements* characteristics, 17 of the data sources contained factors that reflected the importance of requirements characteristics. As a result, the initial reference framework provides for a *Requirements* factor classification. The remaining classifications emerged as a result of applying the Grounded Theory principle of constantly comparing until the major themes become evident.

5.6.3 Phase 3 – Final review

In this final phase, the set of 48 factors and 11 associated classifications that were produced in the constant comparison and memoing phase were examined for possible omissions. In theory, possible omissions can include sub-factors, factors or entire classifications. Therefore, as the constant comparison progressed, part of the memoing exercise involved making notes of factors that were possibly absent. Following the completion of the constant comparison, a detailed examination of these memos revealed that there were six areas that would benefit from extended coverage. These areas are identified in the column labelled *Factors identified in review phase* in Table 4, and they are outlined in more detail in the following paragraphs.

Firstly, in relation to the *Application* classification, the *Deployment Profile* is added as a factor. Furthermore, two additional sub-factors are included with this new factor: *number of deployed versions of applications* and *number of deployed applications*. This addition is considered important as the number of applications and application versions under deployment can have a significant impact on the testing, support and maintenance requirements for a company (as well as the supporting infrastructure, including configuration management and documentation).

		Data Sources																						
Classification	Factors	Ferratt & Mai (2010)	Cameron (2002)	Benaroch & Appari (2010)	Appari & Benaroch (2010)	Wallace & Keil (2004)	Wallace, Keil & Rai (2004)	Ropponen & Lyytinen (2000)	Lyytinen, Mathiassen & Rop (1998)	Keil et al. (1998)	Barki, Rivard & Talbot (1993 & 2001)	Boehm (1991)	Casher (1984)	Boehm CoCoMo (2000)	Albrecht FPA (1984)	Putnam SLIM (1978)	Delany & Cunningham (2008)	Ropponen & Lytinen (2000)	Xu & Ramesh (2007)	SWEBOK (IEEE 2004)	ISO-12207(2008)	CMMI(2006)	Boehm & Turner(2003)	Factors identified in review phase
Personnel	Turnover, Team Size	●				●					●		●	●					●	●				
	Culture	●																	●				●	
	Experience, Cohesion, Skill, Productivity	●	●	●	●	●	●	●		●	●		●	●		●	●		●	●		●	●	●
	Commitment					●																		
	Disharmony					●					●													
Requirements	Changeability, Feasibility	●	●		●	●	●		●	●		●	●						●		●		●	
	Standard, Rigidity		●		●	●	●	●	●	●	●	●	●	●		●	●				●			
Application	Degree of Risk	●	●			●					●		●	●										
	Performance	●		●				●				●		●	●		●							
	Complexity, Type, Size, Predictability, Connectivity			●		●	●	●	●		●			●	●	●	●	●	●	●	●		●	
	Reuse													●	●									●
	Development Phase																			●				
	Deployment Profile																							●
	Quality																			●				
Technology	Knowledge			●	●	●					●			●			●			●				
	Emergent				●	●				●	●		●	●			●							
Organisation	Maturity			●	●			●			●	●		●		●	●	●	●	●				
	Management Commitment					●				●	●						●		●					
	Stability					●	●				●								●					●
	Structure								●															
	Facilities								●												●			
	Size																			●				
Operation	End-Users					●	●			●	●		●				●							
	Prerequisites														●		●					●		
Management	Expertise					●	●	●	●	●	●	●	●	●			●	●	●			●		
	Accomplishment					●					●						●							
	Continuity					●	●										●							
Business	External Dependencies					●		●	●		●	●	●	●			●		●					
	Business Drivers																		●			●		
	Time to Market																			●				
	Customer Satisfaction																			●				
	Payment Arrangements																							●
	Opportunities																				●			
	Magnitude of Potential Loss										●		●											●

Table 4 Classifications, factors and data sources

Secondly, in relation to the *Organisation* classification, the new sub-factor *Rate of organisational change* (*growth or decline*) was added to the *Stability* factor. The rate of growth or decline in an organisation can have a significant impact on the software development process – for example, if a company doubles the headcount from 100 to 200 software engineers over a 12 month period, it is likely that this will have an impact on the required software development process (certainly it will have an impact on the potential economy of scale of the process). Equally, if an organisation reduces its headcount from 200 to 100 engineers, it is probable that there will be an impact on the software development process

Thirdly, in relation to the *Business* classification, the new factor *Payment Arrangements* was added. The payment terms for both on-going and new projects can have an impact on the process definition. If the payment terms are rigidly fixed against contractual requirements and deadlines, then an agile development process (where the requirements are not well understood at the outset) may not represent the best choice of development approach. Equally, if customers want to innovate a new solution with uncertain requirements and a cost reimbursement approach to payment, then a strict traditional waterfall approach may not be well suited to the needs of the project. Therefore, the *Payment Arrangements* are an important situational factor affecting the process definition.

Fourthly, in relation to the *Cohesion* factor, categorised under the *Personnel* classification, two additional sub-factors were added: *Distributed team* and *Team geographically distant*. Although these two new sub-factors are to some extent implicitly covered by other sub-factors such as *Team ability to successfully complete a task*, it is the view of the researcher that the sometimes distributed nature of software development (especially where this is international) is significant enough to be given explicit mention. Where software development teams are distributed beyond a single office, new challenges are introduced, particularly in relation to communication and co-operation. Therefore, these two new sub-factors have been added.

Fifthly, in relation to the *Component Reuse* factor, categorised under the *Application* classification, a new sub-factor was added: *Extent of utilisation of externally-sourced* *components*. For example, with the emergence of the open source development community over recent years, and the increasing probability that cloud computing may be the future for some software development efforts, the use of externally developed software and the management of the supply and maintenance of that software is something that could have an impact on the software development process. A second example of an externally-sourced component includes Commercial Off-The-Shelf (COTS) software.

		Description
Classifications	Personnel	Constitution and characteristics of the non-managerial personnel involved in the software development efforts
	Requirements	Characteristics of the requirements
	Application	Characteristics of the application(s) under development
	Technology	Profile of the technology being used for the software development effort
	Organisation	Profile of the organisation
	Operation	Operational considerations and constraints
	Management	Constitution and characteristics of the development management team
	Business	Strategic and tactical business considerations

Table 5 Classifications and descriptions

Finally, an additional sub-factor, *Loss of human life*, was added to the *Magnitude of potential loss* factor associated with the *Business* classification. While the criticality of the applications under development does consider the importance of the criticality of application in the factor *Type*, it is important to explicitly list the potential for loss of life as a factor for the business to consider.

Having added the six new items listed above, a final classification and factor label review was undertaken to ensure that the overall concepts that emerged in the distillation were accurately and succinctly represented in the final reference framework. This involved a further consolidation of some of the classifications and factors, the results of which were brought forward to an expert review.

The earlier review initiatives have helped to ensure that the reference framework is internally consistent and valid. However, in order to provide an extra level of certainty regarding the complete and consistent nature of the reference framework, an expert review was undertaken. Using the contact network of the researcher, two expert reviewers were secured. In the case of both reviewers, they had extensive

	Factors	Sub-Factors
Personnel	Turnover	Turnover of personnel
	Team Size	(Relative) team size
	Culture	Team culture / Resistance to change
	Experience	General experience / Team experience / Team diversity / Team Ability to understand the human implications of a new information system / Team Ability to work with top management / Application experience / Analyst experience / Programmer experience / Tester experience / Experience with development methodology / Platform experience
	Cohesion	General cohesion / Team members who have not worked for you / Team not having worked together in the past / Team ability to successfully complete a task / Team ability to work with undefined elements and uncertain objectives / Overdependence on team members / Distributed team / Team geographically distant
	Skill	Operational knowledge / Team expertise (task) / Team Ability to work with undefined elements and uncertain objectives / Training development team members / Expectation of personnel's abilities / Analyst capability / Programmer capability / Tester capability / Team understanding of application
	Productivity	Team ability to carry out tasks quickly / General productivity
	Commitment	Commitment to the project among team members
	Disharmony	Interpersonal conflicts
Req's	Changeability	Scope creep / Continually changing system requirements / Ill-defined project goals / Gold plating / Unclear system requirements
	Feasibility	Straining computer-science capabilities
	Standard	General quality of input and output requirements / Conflicting system requirements / Incorrect system requirements / Misunderstanding of the requirements / Engagement of end-users in requirements capture / User understanding of requirements
	Rigidity	Rigidity of compliance to requirements
Application	Degree of Risk	Number of people and departments that the project affects / Thoroughness of design and risk resolution / Application causes major changes to the way end-users work
	Performance	Evaluation of performance requirements / Real-time performance shortfalls / Required reliability / Estimation of hardware/software capabilities
	Complexity	Product complexity / Hardware architecture / Task complexity
	Type	Application type / Application domain / Application criticality / Architecture type / Configuration demands / Back up and recovery demands
	Application Size	Hardware / Software / Required storage / Relative project size and duration
	Predictability	Extent of recent changes / Platform volatility
	Connectivity	Number of links to existing systems / Number of links to future systems
	Reuse	Required reuse / Extent of utilisation of externally-sourced components
	Development Phase	Development / Maintenance / Other phases (e.g. end of life)
	Deployment Profile	Number of deployed versions of applications / Number of deployed applications
	Quality	Required product quality / Maintainability
Tech.	Knowledge	Language experience / Tools experience / Experience with (general) technology / Project involves use of technology that has not been used in prior projects / Introduction of new technology (to general solutions base) / Need for new hardware/software
	Emergent	Emerging technology (the technology itself is emergent)
Organisation	Maturity	Maturity of the organisation / Use of modern programming practices / Availability of technical support
	Management Commitment	Senior management commitment to project / Lack or loss of organisational commitment to project
	Stability	Resources shifting from the project due to changes in organizational priorities / Unstable organizational environment / Affect of corporate politics on projects / Organization undergoing restructuring during the projects / Rate of organisational change (growth or decline)
	Structure	Organisational structure
	Facilities	Physical working arrangements / Facilities to house the project
	Organisation Size	Size of the organisation
Oper.	End-Users	Users resistant to change / End-user commitment / Degree of user engagement with development team / Conflict between users / Conflict between users/departments / Number of users outside the organisation / Number of users in organisation / Number of hierarchical levels occupied by end-users / User turnover / End-user experience with the activities to be supported by the future application / End-user familiarity with application type / End-user understanding of system capabilities and limitations
	Prerequisites	Applicable standards / Applicable laws / Organisational policies / Common practices / Operational ease / Installation ease
Management	Expertise	Effectiveness of project management methodology / Project planning capability / Project management systems / Experience with project management tools / Efficiency of governance structure / Appropriateness of rewards / Project sizing capability / Achievability of schedules and budgets / Estimation capability with respect to the personnel needs of projects / Degree of people skills in project leadership / Progress control capability / Effectiveness of work flow and coordination / Definition of project milestones / Effectiveness of project managers / Management communication skills / Manager familiarity with team / Effective understanding of responsibilities / User expectation management capability
	Accomplishment	Project management experience / Operational knowledge of leader
	Continuity	Changes in organisational management
Business	External Dependencies	Dependency on outside suppliers / Number of hardware suppliers / Number of software suppliers / Multiple implementers / Multisite development / Number of involved parties / Reliance on other projects or processing systems / Number of (external) stakeholders / Stakeholders' background / Access to Stakeholders
	Business Drivers	Project drivers / Finance considerations / Marketing activites / Maximise profit/turnover / Minimise costs
	Time to Market	Time to Market
	Customer Satisfaction	Customer satisfaction / Satisfaction with user interface
	Payment Arrangements	Time and Materials / Fixed price / Non-conventional payment arrangement
	Opportunities	Project opportunities
	Magnitude of Potential Loss	Customer relations / Financial health / Competitive position / Organisational reputation/survival / Market share / Loss of human Life

Table 6 Classifications, factors and sub-factors

SME-based experience across the full spectrum of software development lifecycle activities in multiple organisations, ranging from programming, to technical leading, to management of software projects, to delivery and support of software systems, to client management. Owing to this broad experience, the reviewers were keenly aware of the many factors that can affect the software development process. The reviewers were asked to focus on a number of aspects. Firstly, they were asked to examine the framework for possible absences. Secondly, they were requested to focus on the labelling and classification that was adopted in the framework and to provide feedback regarding the appropriateness of the scheme. Thirdly, the expert reviewers were asked to comment on their opinion regarding the utility of the framework.

Having carefully reviewed the framework, the reviewers made a number of recommendations which were applied to the draft framework. Perhaps the most valuable feedback from the reviewers was a recommendation that each of the 44 factors be given unique labels. In the draft framework, several factor names were reused. For example, there was a *volatility* factor under both the *Requirements* classification and the *Management* classification. Following the expert review, these factor labels were renamed to *Changeability* (in the case of *Requirements*) and *Continuity* (in the case of *Management*). There were a number of other instances of factor name reuse, all of which were addressed through similar relabelling. The introduction of unique factor labels improved the understandability and usability of the reference framework. Furthermore, the use of unique factor labels removed the possibility of confusion associated with using common factor labels to refer to two distinct factors.

The reviewers also recommended that some sub-factors could be further consolidated. For example, the *Potential loss* factor had originally included a *Possibility of lost funds due to system failure* sub-factor which was recommended for consolidation into the existing *Financial health* sub-factor. The reviewers highlighted other instances where such consolidation was recommended and subsequently, the draft framework sub-factors were further consolidated in line with the feedback from the reviewers. The reviewers also suggested that broad community-level collaboration would be a requirement if producing a consensually-agreed general reference framework of the factors affecting the software development process (a point which is discussed in chapter 10 as an area for future work). However, the reviewers reported that the initial reference framework was in their view comprehensive and useful in its present stage of development.

Following the completion of the final review phase, the initial reference framework of situational factors consisted of 8 classifications and 44 factors that inform the software process. This final iteration, along with the association back to the data sources, is summarised in Table 4. The situational factors outlined in Table 4 have a broader listing of 170 sub-factors – these are the components of the factors themselves. These sub-factors have been carefully and diligently carried forward (and consolidated) from the data sources so that a substantial level of detail is available within the reference framework. A description of the 8 classifications is presented in Table 5, with a complete listing of the sub-factors presented in Table 6. In order to provide an easily digestible view of the contents outlined in Table 4, Table 5 and Table 6, the essential components of the reference framework of the situational factors that affect the software process definition are summarised in a consolidated form in Figure 17.

5.7 Survey instrument for examining situational change

Using the situational factors affecting the software development process reference framework, a survey instrument for examining the degree of situational change in an organisation was developed. The primary focus of the survey instrument was to provide a profile of the amount and type of situational change that had occurred over the past year to situational factors that are known to affect the software development process. The guiding principle was that all of the 44 individual factors should be addressed in individual questions in the survey, and where appropriate multiple questions should be developed for and individual factor (say, for example, where there are a large number of sub-factors).

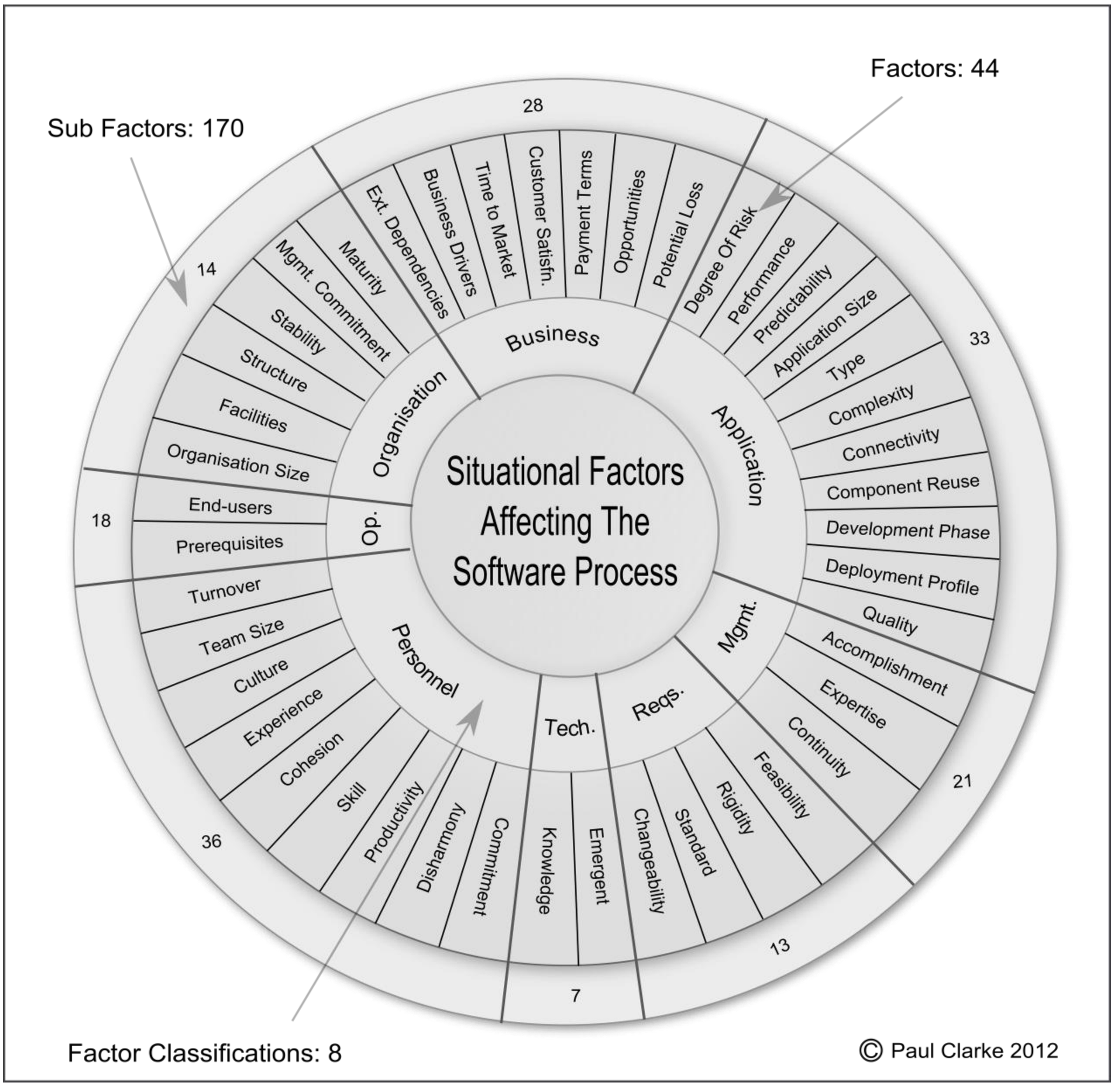


Figure 17 Situational factors affecting the software process

Gradually, a series of questions was developed from the situational factors framework. Since the survey instrument was interested in the amount of change as well as the factors undergoing change, the basic structure of the questions took the form of: *Have there been any modifications to [an aspect of the situation that can affect the software development process]?* By structuring the questions in this way, it was possible to get information on any modification – no matter how large or how small. This approach permits the elicitation of a comprehensive view of the extent and type of situational changes (affecting the software development process) that have manifested in an organisation.

In constructing the survey instrument, the basic classifications and factors that were outlined in the situational factors framework (refer to Figure 17) were preserved.

Therefore, the main body of the survey instrument has eight separate sections, one for each of the classifications in the situational factors framework. Since the classifications had resulted from the extensive constant comparison and memoing applied during the framework construction, it was the view of the researcher that the inclusion of the classifications in the survey instrument would benefit the participants in delineating between different major areas of consideration. This step also permitted the researcher to more easily guide participants through the survey and to provide updates on progress as the survey instrument was discharged.

Many of the sub-factors from the underlying situational factors framework were included in the survey instrument for examining situational change. For example, in relation to the *Prerequisites* factor, the following question was developed: *Over the past year, has there been any modification to the operation prerequisites, including applicable standards and laws?* Through using the examples associated with the questions (for example: *applicable standards)*, the main thrust of the sub-factors is also incorporated into the question. In this way, the fidelity of the underlying situational factors reference framework is significantly retained in the resulting survey instrument. This step was also considered important as it ensured that the significant effort that was applied to developing the situational factors framework was being fully realised in the survey instrument.

As with the SPI activity survey instrument, the situational factors survey instrument was subjected to a pilot with an industry partner. The purpose of the pilot was to check that the survey instrument was fit for purpose and that it could be discharged in a practical fashion. Moreover, the pilot was used to check that the participant could relate to and understand the various questions contained in the survey instrument. At the start of the pilot, the industry partner was notified that it was a pilot-run and they were encouraged to provide feedback on the content, flow and understandability of the survey. Since a great deal of time and effort had gone into the development of the situational factors framework and the associated survey instrument, it is perhaps not surprising that there were few recommendations from the industry partner. The primary item of feedback was a suggestion to repeatedly reiterate throughout the survey discharge that the period under investigation was the previous 12 months. This recommendation was adopted and future participants were continually reminded of the period under investigation.

Regarding the content, flow and purpose of the survey instrument, the industry partner was very positive concerning the general experience, and felt that the survey instrument provided an interesting mechanism for examining the situational changes that affect the software development process. The pilot was the final phase in the survey instrument creation, after which the finished form of the survey instrument was evident. In total, 49 individual questions for examining the degree of situational change are contained in the survey instrument, which is presented in full in Appendix C.

5.8 Summary

This chapter presented the approach adopted when collecting data in relation to the second major data component of the study: the degree of situational change. In order to examine the degree of situational change in a software development setting, it is necessary to have a reference framework of the factors of the situational context that affect the software development process. No such framework had previously been published and therefore, this chapter outlined the approach taken in constructing an initial reference framework of the situational factors affecting the software development process. Finally, the reference framework was adopted for the construction of a survey instrument that can be discharged in order to determine the degree of situational change in a software development company.

In the next chapter, the approach adopted when collecting data in relation to the third and final major data component is presented: the extent of business success.

6 Examining the Extent of Business Success

6.1 Introduction

This chapter outlines the approach to examining the extent of business success in the participating organisations. The initial part of this chapter is dedicated to examining the meaning of success in the general sense, for an organisation of any type. The classical and contemporary views of business success are outlined and examined, demonstrating that both financial and non-financial views of business success are considered to be important. The latter part of this chapter outlines the different perspectives of business success that are specifically considered important for software development companies, with the final section outlining the business success survey development technique.

6.2 Measuring business performance

In the business literature, the term *success* is used interchangeably with the term *performance*, with both terms representing the achievement of something desired, planned or attempted (Maidique and Zirger 1985). However, beyond this general description, controversy exists in relation to what exactly is meant and understood by the term business performance (Morgan and Strong 2003). Businesses measure performance for a variety of different reasons including: the identification of improvement opportunities, determinations in relation to customer satisfaction, to enhance understanding of their own processes and to assess the degree of success achieved (Parker 2000). This variety of reasons for measuring performance has given rise to a variety of different performance measures that can be classified into one of two groups: financial and non-financial (Hart 1993).

6.2.1 Financial measures of performance

Traditionally, business performance has been measured in purely financial or accounting terms (Jennings and Seaman 1994). Profitability, usually measured by return on investment (ROI), has by convention, been used to assess performance and is widely regarded as the ultimate bottom line test of success (Morgan and Strong 2003). In addition to ROI, other financial measures of business performance include return on sales, sales per employee, productivity and profit per unit production (Ghalayini and Noble 1996).

The financial perspective has been reported as having a significant impact on performance – with Reid and Smith (2000) concluding that the pursuit of the highest rate of ROI is a primary consideration for owners and managers. This view is long established in the business success domain with Ansoff (1965) asserting that "*return on investment is a commonly and widely accepted yardstick for measuring business success*" (Ansoff 1965 p42).

6.2.2 Obtaining financial data

For a researcher, the collection of financial data from small and privately held firms poses significant challenges. One concern relates to the reluctance of businesses to report their financial position (Kotey and Meredith 1997). This issue is also observed by Hart (1993) who highlights the "*reluctance and inability [of firms] to part with financial data*" (Hart 1993 p30) and again by Dess and Robinson (1984) who assert that owners can be very sensitive in relation to releasing financial performance data. In addition, Dess and Robinson (1984) acknowledge that Hollander's concept of *idiosyncratic credits* (Hollander 1974) is relevant to privately-held firm research: a researcher may accumulate credits with a business owner through expertise and interest only to consume these credits in obtaining meeting time and specific information. The risk is that a request for sensitive performance data may exhaust or exceed the entire fund of credits.

Dess and Robinson (1984) examined the possibility of using subjective data as opposed to concrete financial data, noting that although objective measures are preferred, "*a researcher might consider using a subjective perceptual measure*" (Dess and Robinson 1984 p271). Therefore, the possibility of using subjective data, garnered from senior managers and owners has been shown to be a valid approach to financial data collection. However, obtaining and utilising actual financial data is considered to be a stronger approach to examining financial success.

6.2.3 Limitations of financial performance measurement

While financial return is an important indicator of business success, it has been established that "*profits are not necessarily the sole purpose of a firm*" (Nonaka and Toyama 2005 p420). Furthermore, it has also been observed that financial return is far from being the only important measure (Maidique and Zirger 1985), with claims that short term financial measures of performance that emphasise a quick return on investment can come at a cost to long term growth (Hayes and Abernathy 1980). Financial measurement can be considered as tangible evidence of performance but other important performance measures should also be assessed so as to prevent the "*inadequate handling of intangibles*" and the "*improper valuation of sources of competitive advantage*" (Bharadwaj, Varadarajan and Fahy 1993 p87). The measurement of customer satisfaction demonstrates the importance of intangible measures and highlights the danger of focusing solely on financial data: a company that posts successful financial returns might appear to be performing well but if all of the clients are dissatisfied, the future profitability prospects for the company will be at risk. As a result of the shortcomings of purely financial performance measurement, there has been a "*shift from treating financial measures as the foundation for performance measurement to treating them as one among a broader set of measures*" (Eccles 1991 p131) and this has given rise to multidimensional performance measurement frameworks.

6.2.4 Multidimensional performance measurement frameworks

Owing to the dissatisfaction with traditional accounting-based performance measurement systems, multi-dimensional performance measurement frameworks were created as an alternative approach to business performance measurement (Bourne et al. 2000). As well as accommodating established financial measures of success, these new frameworks incorporated non-financial, future looking performance measures.

A number of multidimensional performance management frameworks have been created, each trying to unlock the vital measurements that would best provide a complete view of the business performance. The *Performance Pyramid* (Lynch and Cross 1990) contains a pyramid of measures aimed at integrating performance through the hierarchy of the organisation. The *Macro Process Model* (Brown 1996) identifies links between the five stages in a business process (inputs, processing system, outputs, outcomes and goals), arguing that each stage is the driver of the performance of the next. *Kanji's Business Scorecard* (KBS) defines four fundamental dimensions to be managed and measured: organisational value, process excellence, organisational learning and stakeholder delight, while the *Performance Prism* (Neely, Adams and Kennerley 2002) consists of five interrelated perspectives: stakeholder contribution, stakeholder satisfaction, strategies, processes and capabilities. However, it is the Balanced Scorecard (BSC) (Kaplan and Norton 1992) approach that is the most popular multidimensional performance measurement framework (Kennerley and Neely 2002) and which has exercised the most influence in the domain of performance management (De Waal 2003).

Following a one year multi-company study in 1990, the BSC approach to performance management was developed with the aim of presenting management teams with a concise summary of the business Key Performance Indicators (KPIs). By facilitating the alignment of business operations with overall business strategy, the BSC provides a mechanism for translating the vision of a company into a set of objectives. In addition to the traditional financial performance measures adopted by companies, the BSC creators also wanted to capture performance ratings for customer satisfaction, internal business processes and the ability to learn and grow (Fernandes, Raja and Whalley 2006). Consequently, the BSC identified four measurement perspectives: *financial*, *customer*, *internal business processes* and *learning and growth*. While the BSC has presented a packaged performance measurement approach that considers four key perspectives that are considered to offer "*good coverage of the dimensions of performance*" (Hudson, Smart and Bourne 2001 p1102), the novelty of the approach has been questioned, with claims that similar multidimensional approaches have existed since at least the 1960s (Pandey 2005).

The *financial* perspective of the BSC is concerned with the sustained growth of shareholder value while the *customer* aspect examines the value for clients. The *internal business process* perspective of the BSC measures internal process quality and *learning and growth* assesses the capacity of the organisation for innovation and continuous improvement. Used together, these four perspectives provide a *balanced* assessment of business performance. The BSC technique involves the determination of strategic objectives for each perspective, obtained by interviewing managers, and the development of specific measurements for the objectives.

6.2.5 Using multidimensional scorecards in SMEs

While there is evidence of the BSC approach being adopted in large companies (Silk 1998), it has been claimed that small companies that are restrained by resources and financing may not be suited to performance measurement using a scorecard type approach – as it can require "*a great amount of time and resources to keep the scorecard updated and effective*" (Gautreau and Kleiner 2001 p154). Moreover, the BSC approach is acknowledged as being difficult to implement, typically taking five or six months to initialise, and a number of additional months to fine tune (Gautreau and Kleiner 2001) – a view supported by McKenzie and Shilling (1998) and Chow, Haddad and Williamson (1997). The BSC has also been criticised for having a lack of consideration for the value of employees and for the importance of knowledge management (Chen, Zhu and Xie 2004).

Although some research has criticised the BSC as being difficult to implement and potentially not suited to small companies, other studies have asserted that the BSC is a credible and valuable approach to performance measurement for SMEs. In particular, it has been noted that SMEs can derive benefits from the BSC approach without having to implement an administratively demanding measurement regime and that SMEs obtain the most value from the BSC when it is used as a mental or verbal frame of reference for addressing general strategic issues and business goals (Andersen, Cobbold and Lawrie 2001). Therefore, rather than implementing the BSC model in its entirety, SMEs should consider using it as a reference model for the definition and tracking of business objectives.

Irrespective of the arguments for and against BSC implementation, it is reported to be the most widely adopted (ADB 2007, Lim and Lee 2008, Barnes and Hinton 2008) and most notable (Tapanya 2004) performance measurement framework.

6.2.6 Performance measurement for software development companies

While the BSC approach could be applied to any business type, the software development business, often characterised by high levels of dynamism and uncertainty, requires a broader approach to performance measurement (Sureshchandar and Leisten 2005). Consequently, Sureshchandar and Leisten (2005) have adapted the BSC approach, rendering a strategic performance measurement and management framework for the software development industry, the Holistic Scorecard (HSC). The HSC comprises of six perspectives: *financial, customer*, *business process, intellectual capital*, *employee* and *social*. While the initial three perspectives are broadly similar to the BSC, the latter three – intellectual capital, employee and social – are new considerations and they reflect some of the key items that may affect the performance of a software business.

Intellectual capital represents intangible assets which frequently do not appear on the balance sheet, and is defined as the difference between the book value of the company and the amount that someone is prepared to pay for it (Brooking 1997). According to Brooking (1997), there are four categories of assets that constitute intellectual capital: assets which give a company power in the marketplace, such as trademarks and customer loyalty; assets representing property of the mind, such as intellectual property in the form of patents and trademarks; assets that give the organisation internal strength, such as culture and business processes; and assets derived from people who work in the organisation, such as individual knowledge and know-how. For software development organisations, especially those that are product development oriented, intellectual capital can represent a significant proportion of the estimated value of the business.

The employee perspective of the HSC recognises the special importance of the workforce to the software development business, where "*the availability and optimum utilisation of competent people, coupled with providing a conducive environment for their professional growth and satisfaction, are key for the survival of software organisations*" (Sureshchandar and Leisten 2005 p10). Managed properly, the human resource of a software development organisation is a potential source of competitive advantage. The social perspective is also important to software development businesses, but is somewhat more abstract than the other perspectives presented in the HSC. By demonstrating a social conscience, through the fulfilment of social and community obligations, an organisation can develop a good image and goodwill, which will improve a customer's overall evaluation of the organisation. Such social and community contributions include educational and career counselling, employment opportunities for the less privileged and welfare activities. In addition to these generic contributions, in order to gain social credits, software development companies need to take care to avoid negative impacts on society in general, such as product failures leading to costly losses.

The HSC is a software development focused extension of the dominant generic business performance measurement framework, the BSC, and it outlines a targeted technique for measuring the performance of software development companies. As indicated by Andersen, Cobbold and Lawrie (2001), such balanced scorecard-based approaches are beneficial for SMEs when implemented in a light-weight fashion that supports the definition and measurement of strategic business goals.

6.3 Survey instrument for examining business success

As outlined in the previous section, the HSC presents a multidimensional approach to examining business success in software development companies. Furthermore, the HSC can be applied in a light-weight fashion in order to identify the objectives for an organisation. In this section, the HSC is outlined in more detail and the approach to transforming the HSC into a survey instrument is also presented.

6.3.1 Transforming the HSC into a business success survey instrument

In addition to the six primary HSC dimensions of success identified in the previous section, the HSC also sets out sixteen Critical Success Factors (CSFs) and a series of sample KPI measures that can be used when examining business success in software development organisations (refer to Figure 18).

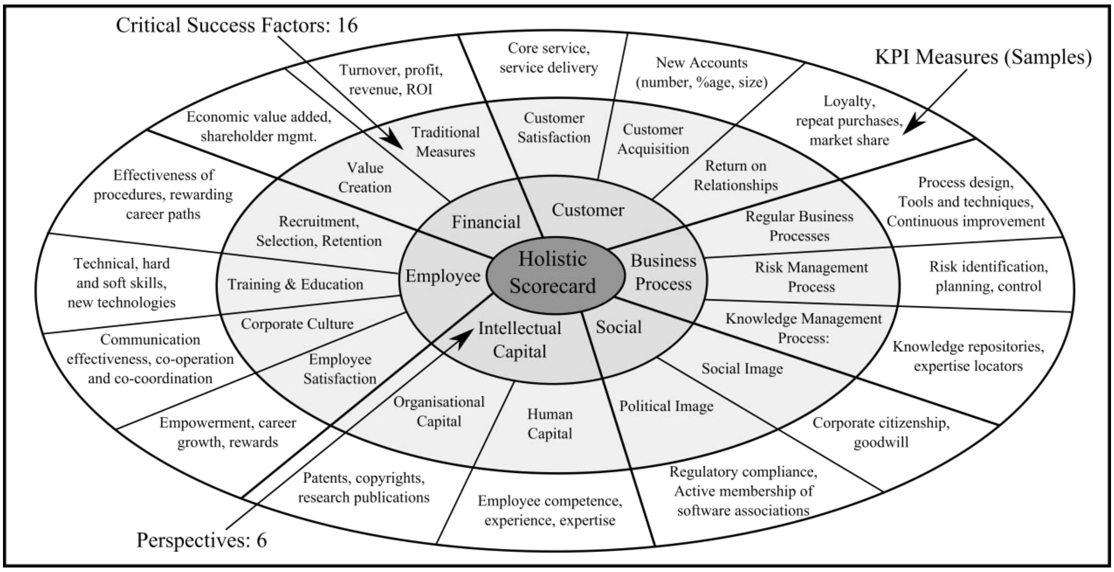


Figure 18 Holistic scorecard

While the HSC identifies a broad spectrum of the performance parameters that are important for software development companies, it does not provide a survey-based instrument for identifying and measuring these parameters. Neither does the HSC offer guidelines on how to reliably collect the business objectives data. Therefore, using the HSC as a reference, an SME business success survey instrument was constructed. As with the earlier survey instrument development (for SPI activity and situational change), care was taken to retain the structure and fidelity of the underlying reference framework when creating the associated survey instrument. This step ensures that the maximum possible value is obtained when discharging the survey instrument. Consequently, each of the six perspectives and the sixteen CSFs are identifiable in the resulting survey instrument – such that it is clear that the survey instrument has been derived from the HSC.

Questions in the business success survey instrument take the general form of: "F*or the forthcoming year, list any objectives that exist in relation to [a business objective]*". Where objectives were identified, the participant further identified if the objective was high or low priority. As described in more detail in chapter 7, the priority is an important consideration for business objectives – as the achievement of higher priority objectives is generally more desirable than the achievement of lower priority objectives.

The survey instrument is deployed in two phases (refer to Figure 19). Firstly, it is utilised to determine the business objectives for the forthcoming period. Secondly, it is deployed in order to determine the extent to which the objectives were achieved. This two-phased approach to determining the extent of business success helps to ensure that the reported success in achieving business objectives is free from biased or false recollections – it also helps to formally identify the objectives in settings where no such formal description may exist. Additionally, a series of questions are added to the survey instrument to support the disclosure of objectives that are beyond the scope of the HSC framework, as encouraged by the creators of the HSC.

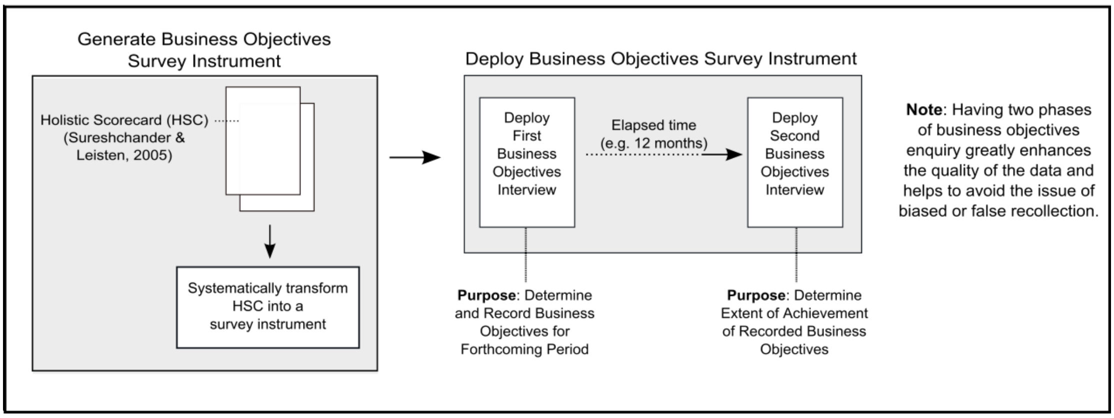


Figure 19 Two phased approach to determining the extent of business success

Although the HSC is a recognised framework for assessing business success in software developed companies and while every care was taken in transforming the HSC into a survey instrument suited to examining business success in software SMEs, it was nonetheless considered prudent to conduct a pilot with an industry partner. The purpose of the pilot was to ensure that the survey instrument was fit-for-purpose, that participants would understand the questions and that the required information would be accessible using the instrument. As with the earlier piloting efforts, the industry partner was made aware that they were participating in the pilot run of the survey instrument and they were encouraged to provide feedback that would benefit future evolutions of the survey instrument.

Overall, the pilot feedback was very positive. One minor area of criticism related to the use of two separate questions for *employee expertise* and *employee competence* objectives for the forthcoming year. The participant reported that it was difficult to distinguish between competence and expertise. As a result, employee competence and expertise were merged into a single question (with both words retained). It also became evident during the pilot that the participating organisation had headcount goals, and as a result, an additional question was added to the survey instrument to identify headcount objectives in future surveys. It is worth noting that the pilot participant was not previously aware of the HSC, which was the case for all subsequent surveys as well. Therefore, participating in the study also brought immediate benefits for the participating organisations, since they became aware of the existence of the HSC and in many cases they remarked that it was an interesting and comprehensive framework for considering business objectives in software companies.

Following the pilot phase, a final rendering of the survey instrument comprising of 51 individual questions for the examination of business success in software development companies was produced. The final version of the business objectives survey instrument is presented in Appendix D.

6.4 Limitations of objective-based success measures

The two-phased HSC-based approach adopted by this research in order to make determinations in relation to business success is robust, representative, and holistic. However, it is just one view of business success and it does suffer from a number of limitations that are explicitly outlined in this section. Firstly, to pursue the optimum success strategy, a business may change its original objectives should better or different or unexpected opportunities arise. As a result, the initial objectives may not be satisfied – but more favourable outcomes may have been achieved. It is also possible that this type of objective volatility could be more prevalent in smaller organisations. Therefore, in cases where the objectives change radically throughout the period under investigation, the assessment of success relative to outdated objectives could be misleading. Although evidence of volatile business objectives was not detected in this study, such characteristics may impact on the efficacy of the two-phased objective-based approach.

A second limitation of objective-based business success measurement is that it lacks an independent evaluation of the extent of business success. This is problematic as two different businesses can set targets with varying degrees of ambition. One company may want to expand rapidly while a second company may be satisfied with the current business landscape. In this situation, the second company may easily achieve its business targets while the first company – though witnessing considerably greater business expansion – may fail to fully achieve its own ambitious objectives. In this instance, using objective-based success measures, the first organisation may appear to be less successful than the second company; resulting in a possibly misleading representation of business success.

Despite the noted limitations of the two-phased approach, it does present as one view of business success that is valid, thorough and useful. Consequently, the HSC-based two-phased business success framework outlined in this chapter is adopted in this research.

6.5 Summary

This chapter presented the approach adopted when collecting data in relation to the third major data component of the study: the extent of business success. In order to examine the extent of business success in a software company, a suitable reference framework of the dimensions of success for software development companies is required. The HSC presents an extensive multidimensional reference framework of the important business success perspectives for software development organisations. The latter part of this chapter outlined how the HSC was fashioned into a survey instrument that can be used to examine the extent of business success. To ensure the maximum validity of the business success investigations, a two-phased approach to business success determination is outlined. The first phase formally captures the business objectives data for the forthcoming period, with the second phase later examining the extent of achievement of the goals that were recorded in the first phase.

In this chapter, the approach to examining the extent of business success was outlined. In the two previous chapters, the approach to examining the extent of SPI activity and the degree of situational change were presented. In the following chapter, the timeline for performing the data collection is outlined, along with a profile of the participating organisations and individuals.

7 Data Collection

7.1 Introduction

Prior to presenting the details of the data collection associated with the study, it is worth briefly recapping on the underlying purpose of the research and the associated data requirements. This research is concerned with examining the relationship between SPI, situational change and business success. In order to investigate the interplay between these three distinct phenomena, it is first necessary to formulate a robust and reliable approach to making determinations in relation to each phenomenon. As outlined in chapter 3, the use of three separate survey instruments, one for each phenomenon, is a desirable and appropriate approach for collecting the required data. Chapters 4, 5 and 6 were dedicated to the development of three survey instruments, one each for investigating SPI activity, situational change, and business success.

The present chapter is concerned with reporting details of the application of the three survey instruments to the task of collecting data from a number of participating software SMEs. Firstly, this chapter discusses the approach adopted when accessing participating companies, along with a description of the confidentiality and privacy measures employed during the data collection. Thereafter, the individual companies and participants are profiled, followed by details of the dates and durations for the data collection. Finally, the actual data collected is presented.

7.2 Accessing organisations

Owing to the time required to discharge the survey instruments and the senior profile of the required participants, one of the major challenges facing this research was the identification and participation of software SMEs. As noted earlier in this thesis, the researcher accumulated thirteen years experience in industry, a fact that was instrumental in securing the participation of study organisations. Using a network of established peers and colleagues from industry, the researcher embarked upon the process of gaining the trust and willing participation of participants.

Initially, a total of 17 software SMEs agreed to participate in the research. Later, two of these companies withdrew from the research, citing business pressures as a barrier to participation. Since this study examines evidence of the relationship between SPI, situational change, and business success, the generalisability of findings dictated that it was vitally important that as many companies as possible (and practicable) should participate in the study. That 15 software SMEs had been secured for participation in the study was generally viewed in a very positive light – especially in consideration of the time requirements for the participants. Furthermore, and as discussed elsewhere in this thesis (chapter 3, chapter 8 and chapter 9), this research finds that 15 organisations were sufficient to develop an indicative insight into the relationship between the three phenomena under study.

There are two major hurdles that need to be navigated in order to secure the participation of companies in studies such as this one: trust and time. Without having secured the trust of the participants, it would not be possible to convince them to share confidential company data (such as financial targets, and planned product development). There are, of course, significant benefits to be derived from working in an environment of respect and trust, especially when it comes to the candid and honest nature of feedback from participants. Even where trust is establishable, the time demands for studies such as this one could act to rule organisations out of the research. The difficulty associated with overcoming the trust and time commitment challenges may well account for the absence of similar studies in the past.

7.3 Confidentiality and privacy considerations

In order to support the elicitation of data and to promote trust, each of the participating organisations was assured that their data would at all times be accorded the highest possible level of confidentiality. As evidence of this commitment to confidentiality, each of the participating organisations was allotted a pseudonym and individuals were identified by job title rather than by name. As a result of these steps, no organisation is identifiable through the material published in relation to the study.

In addition to the allocation of pseudonyms to the participating organisations, a number of extra steps were employed in order to safeguard the confidentiality of the participating companies, as follows:

- Where deemed appropriate, a bi-lateral non-disclosure agreement (NDA) was established between the research team and the participating organisation.
- All interview recordings and transcriptions were encrypted – this encryption was applied to both the primary and the backup media, and includes data stored on portable recording devices.
- At no stage would the participating organisations be identified, either in documented form or in verbal discussion, to a third party.

Once informed of these confidentiality measures, the participants felt more secure in their contribution, safe in the knowledge that it was being accorded a formal and very high level of confidentiality and security.

7.4 Profile of participating organisations

The study examined 15 distinct organisations, with the largest recorded headcount being 120, and the lowest recorded headcount being 4. In addition, none of participating organisations had an annual turnover exceeding 50 million euro and/or an annual balance sheet total exceeding 43 million euro. Therefore, all of the participating organisations qualify as SMEs under the European Commission's definition (European Commission 2003). While the majority of the organisations in the study group retained their head office in the Republic of Ireland, all of the participating organisations deliver products to the global market place. Two of the companies were predominately based outside the Republic of Ireland, with a further four organisations retaining either development or operational centres internationally. Three of the participating companies had less than 10 staff, while 3 of the companies had between 10 and 19 staff. The remainder of the participating organisations had between 20 and 120 staff.[3]

The participating companies operate in diverse business domains. Four of the organisations develop web-based software, with another four organisations

[3] These staff headcount statistics are extracted from the data collected at the end of the period under study.

developing software for the telecommunications domain. The remainder of the organisations operate in a variety of different sectors, including, content management, data mediation, and embedded software. The diverse nature of the business domains was considered important to the study, as it further supported the broader applicability that could be reflected in the findings. Had the participating organisations been drawn from one business domain alone, the value of the findings to other domains could be diminished.

7.5 Profile of participating individuals

A further challenge facing this research related to the nature of the data that was required. The collection of SPI data was based on the broad spectrum of processes in ISO/IEC 12207, and therefore it was necessary to interview personnel who would be aware of changes across a broad expanse of inquiry. As a result, only very few individuals (and sometimes only a single individual) in the participating organisations were able to provide the information required. This issue also arose in the case of the situational change and business success. Therefore, particular individuals needed to be targeted in the participating companies – and often, these individuals were in positions of intense day-to-day pressure, hence increasing the difficulty of accessing the required data. In this respect, it was not uncommon for scheduled meetings to be postponed on many occasions, and sometimes, frustratingly, at the last minute. However, perseverance and patience tended to overcome these challenges, and as noted earlier, of 17 companies that originally agreed to participate at the study outset, only two companies later withdrew prior to study completion.

In some cases (predominately the smaller organisations), it was desirable to consult just one individual person in order to gather the data in relation to all areas of investigation. However, in other cases, it was necessary to consult more than one individual in a participating firm. While it is logistically challenging and generally more difficult to secure access to multiple personnel in an organisation, there are significant benefits to this type of inquiry. Firstly, since more than one point of view is taken into account, the resultant data can be considered to be more broadly founded and potentially more accurate. Secondly, by getting access to the most informed personnel for a particular consideration, it is possible to get a closer approximation to the truth. This greatly enhances the overall quality of the data with respect to examining the hypotheses and research questions set out in this study.

In general, the survey questions regarding business success were addressed by the Chief Executive Officer (CEO), the Chief Operating Officer (COO), the Managing Director (MD), or the Director of Finance (DF). Questions in relation to the amount of SPI activity and the degree of situational change were generally answerable by the Director of Engineering (DE), the Chief Technical Officer (CTO), the Engineering Manager (EM) or the Development Manager (DM). With job titles varying from organisation to organisation (as does the remit and extent of knowledge of the individual undertaking any particular role), one of the initial discussions with each of the participating organisations was aimed at identifying the suitable person(s) to participate in different aspects of the investigation. A complete listing of the study participants (by role and company pseudonym) is provided in Table 7.[4]

Company Pseudonym	Business Success	SPI Activity	Situational Change
Silverback	CTO	CTO	CTO
Grenoble	CEO	EM	CEO, EM
Mega	MD	MD	MD
Cameron	MD	DM	MD, DM
Colleran	CEO	CEO	CEO
Lakes	MD	CTO	MD, CTO
United	MD	MD	MD
Watch	DF, CTO, DE	DE	DF, DE
BocaJ	MD	MD	MD
Tribal	DE	DE	DE
Dynamic	DE	DE	DE
Michelin	DE	DM	DE, DM
LordHenry	DE	DE	DE
When	COO	COO	COO
Oryx	COO	DM	COO, DM

Table 7 Participating organisations and personnel

[4] In order to safeguard the anonymity of the participating organisations, complete company profiles are not presented herein. This is considered to be of high importance since (1) a number of the companies operate in direct competition, and (2) this thesis will be subject to public consumption.

7.6 Data collection periods and durations

Using the survey instruments outlined in chapters 4, 5 and 6, the data collection associated with the investigation into the relationship between business success, SPI activity and situational change was discharged over a 16 month period. A total of four separate engagements are required with each company: one each for SPI activity and situational change, and a further two for business success (as presented in detail in chapter 6). In the case of business success, it is necessary to establish the business objectives for the year in advance, later returning to the organisation to determine the extent to which the objectives were achieved. As a result, the business success data was collected at two separate points in time.

Initially, the participating organisations were engaged in the period March-July 2010 so as to (1) identify their objectives for the forthcoming year. A year later, in the period March-July 2011, the organisations were revisited, at which stage the focus was on determining: (2) the extent to which the recorded objectives were achieved; (3) the amount of SPI activity undertaken during the preceding 12 month period; and (4) the degree of situational change evident in the organisation over the preceding 12 month period. An overview of the timeline for engaging with the participating organisations is presented in Figure 20.

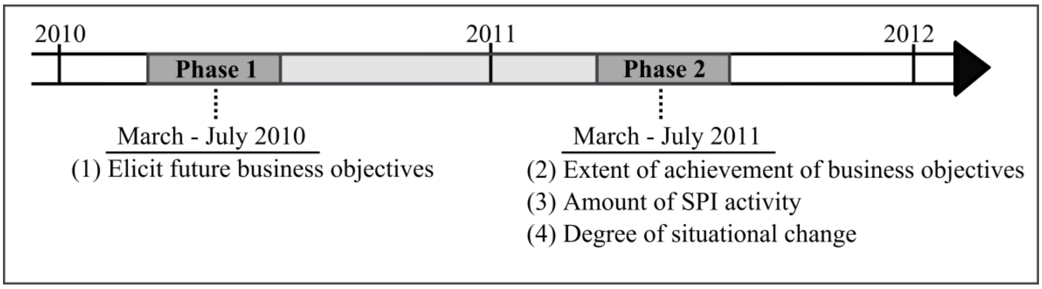


Figure 20 Time periods for data collection

With two business success interviews required for each of the fifteen participating software SMEs, and a further two interviews for SPI activity and situational change, a total of sixty distinct interviews were completed. On average, the total time to interview with each organisation was approximately 5.5 hours, giving a total interview time of approximately 82.5 hours. For the most part, the interviews were conducted face-to-face; however, in some cases (for example, where the interviewee was based internationally), the interviews were conducted by telephone or videoconference. Each of the initial business objectives interviews took approximately 1.5 hours to discharge, with the follow up business success interviews requiring on average 45 minutes. The SPI activity interview required on average 2 hours and the degree of situational change interview required approximately 1.25 hours.

7.7 Collected data

The collection of data from the participating individuals was achieved using the survey instruments which were presented in chapters 4, 5 and 6 (with the actual survey instruments presented in full in Appendixes B, C & D). When conducting the interviews, the primary consideration was to agree a rating with the participant regarding the questions in the survey instrument. Often, some discussion would occur when agreeing the rating, with such discussions providing additional interesting insights. Where such insights and comments arose, they were carefully collected and documented along with the interview records. While the ratings permitted a numerical or statistical analysis of the data, the commentary often afforded additional supporting (and sometimes entirely new) information. The collection of both commentary and numerical rating values is a feature of all field data collection for the study, with the collection of both types of data facilitating a thorough and in-depth assessment of the phenomena under investigation.

The first field engagements were concerned with the collection of data regarding the business objectives for the forthcoming year. The objective ratings collected during the first engagement indicate the importance of each stated business objective as agreed with the participant (as per the Likert scale presented in Table 8). The resulting interview records include details of the business objectives for the forthcoming year, their ratings, and additional comments. The consolidated view of the business objective ratings is presented in Table 9.

Rating Value	Rating Interpretation
0	No objective exists
1	A low priority objective exists, but with no explicit target
2	A low priority objective exists, with an explicit target
3	A high priority objective exists, but with no explicit target
4	A high priority objective exists, with an explicit target

Table 8 Business objective rating scale

The business objectives rating scale could have been limited to just three points: *high priority, low priority*, and *no objective*. However, it was felt that objectives which had an associated formal target were (1) more easily quantifiable in terms of achievement, and (2), were of a perceived higher importance (since the organisation had gone to the effort of explicitly setting a performance target). Therefore, a five point scale was adopted, within which it was possible to identify objectives that had explicit performance targets. For example, a high priority objective to grow the year-on-year profit by 15% will have a rating value of 4, while a low priority objective to increase the number of customers but with no explicit target will have a rating value of 1. Capturing the objectives in this manner permits a later detailed examination of the extent to which the stated objectives were achieved.

Once the 12 month period under investigation elapsed, each organisation was revisited, wherein each of the recorded business objectives (from the first interview) was rated in terms of achievement. The achievement of each business objective was rated according to the Likert scale identified in Table 10, with the consolidated view of the objective achievement ratings for each organisation presented in Table 11.

For the SPI activity and situational change investigations, a similar approach to data collection was adopted. The survey instruments identified in chapters 4 and 5 were applied to the task of examining the amount of SPI activity and the extent of situational change. For each of the questions in the survey instruments, a rating was agreed with the participant, as per the Likert scale presented in Table 12. In addition to identifying the modification ratings for the various different aspects of the software development process and the situational factors, any additional interesting commentary from the participants is once again noted. The overall view of the SPI activity and situational change is presented in Table 13 and Table 14 respectively.

Objective	Strength of Objective														
	SilverB	Michelin	Grenoble	When	Tribal	BocaJ	Watch	LordHenry	Lakes	Colleran	Cameron	Mega	United	Oryx	Dynamic
Revenue	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Profit	4	4	4	4	4	4	4	3	4	4	4	2	3	4	4
Number of new client acquisitions	4	3	4	4	4	4	4	0	4	4	4	0	4	4	3
Extending product offering with new product features/offerings	1	1	4	4	3	3	4	4	3	3	3	3	4	4	4
Number of existing clients presenting for repeat business	3	1	4	4	3	4	3	1	4	4	4	4	4	3	1
Business process management	1	2	1	3	1	2	3	4	4	1	3	4	3	3	3
Customer satisfaction (make the customers more or less satisfied)	3	4	3	3	1	1	1	2	3	1	1	0	3	3	0
Profile of new client acquisitions (larger, smaller, same)	2	3	3	3	3	2	1	0	2	1	0	0	3	4	2
Compatibility with regulatory bodies and local government	0	1	0	3	4	0	0	4	3	3	1	0	0	3	3
Brand image	3	1	1	0	1	3	1	1	1	1	4	2	1	3	1
Changing the nature of customer interaction	0	3	1	1	1	0	0	4	4	0	0	3	0	3	1
Organisational productivity	1	3	1	1	1	1	1	1	1	3	1	0	4	1	3
Employee competence & expertise	2	1	1	2	1	0	1	2	1	1	1	2	1	2	1
Market share in core and allied markets (%age growth or decline)	1	1	2	2	1	4	1	1	0	1	2	0	0	1	1
Pay and perks	2	1	0	2	2	0	2	0	0	0	2	2	0	2	2
Focus on continuous improvement or benchmarking against best practice	1	1	1	3	1	1	1	1	1	1	1	0	2	1	1
Employee Experience	1	2	2	2	1	0	1	1	1	1	1	0	1	2	1
Aid and subsidies, infrastructural support, tax exemptions, etc., from gov.	0	0	0	2	2	1	2	2	0	1	2	0	1	2	1
Skill sets	1	0	2	1	1	1	1	2	0	1	0	2	1	2	1
Organisational Headcount	0	1	2	2	1	0	0	1	2	1	1	2	1	1	1
Retention strategies (especially for precious resources)	2	1	1	1	0	0	0	0	0	1	2	1	2	1	2
Communication effectiveness	0	2	1	1	1	0	0	2	1	1	1	1	1	2	2
Effectiveness of business tools and techniques	1	0	0	1	1	1	1	1	1	1	1	1	0	2	1
Developing or extending internal knowledge repositories	0	1	0	3	3	0	1	2	1	0	0	1	1	0	1
The effectiveness of recruitment and selection procedures	1	1	1	1	1	0	1	1	1	1	1	0	1	2	1
Employee empowerment	1	1	0	1	0	0	1	1	1	3	1	1	2	1	1
The diffusion of knowledge throughout the organisation	0	1	0	2	1	0	1	2	1	0	1	0	2	1	1
Patents, copyrights, branding and trade secrets	0	1	2	2	0	0	0	1	0	2	2	0	0	0	2
Co-operation among people at different levels, functions and departments	0	1	1	1	2	0	1	2	1	1	0	0	2	1	0
Structured and rewarding career paths	1	1	1	1	1	0	1	1	0	1	1	0	0	1	1
Attitude To Work	1	1	1	2	1	0	1	1	1	1	1	0	0	0	2
Training in technical, hard and soft skills, and new technologies	0	1	0	2	1	2	1	0	0	1	1	0	0	0	1
Effectiveness of business documentation	0	1	0	1	0	1	0	1	1	2	0	0	1	1	0
Risk management planning	1	0	2	0	0	0	1	1	0	1	1	1	0	1	0
Career growth	1	0	0	1	1	0	1	0	0	1	1	0	1	1	1
Other specific financial measure	0	0	0	0	0	0	0	0	0	3	0	0	4	0	0
Employee qualifications	0	1	0	1	1	0	0	2	0	0	0	1	1	1	0
Active membership in s/w assoc'ns, instit'ns, soc's and communities	1	1	0	0	0	0	1	1	0	0	1	0	0	1	1
Contribution to society (e.g. job opportunities for the less privileged)	0	0	0	0	0	1	0	0	1	1	1	1	1	1	0
Intellectual Agility	0	0	0	0	0	0	0	0	0	0	0	1	0	2	2
Corporate citizenship (self-regulation wrt legal and ethical standards)	0	1	0	0	0	0	0	1	1	1	1	0	0	1	0
Seeking / Retaining a recognised quality standard (e.g. ISO)	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
Rewards / Recognitions	0	1	0	1	0	0	0	0	0	1	0	0	0	0	2
Goodwill, general image among the common public	0	0	0	0	0	1	0	0	1	1	1	0	0	1	0
Registered designs and processes	0	1	0	0	0	0	0	0	0	2	1	0	0	0	0
Research publications	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
The alignment of training programmes with long term organisational goals	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Training effectiveness	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Change management (i.e. chaperone employees through change events)	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Assets (not including intellectual capital)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Redressing grievances	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 9 Objective rating per objective per organisation

Achievement Value	Achievement Interpretation
0	Not achieved to any extent
1	Partially achieved
2	Mostly achieved
3	Totally achieved

Table 10 Achievement rating scale for business objectives

Objective	Extent of Achievement														
	SilverB	Michelin	Grenoble	When	Tribal	Bocal	Watch	LordHenry	Lakes	Colleran	Cameron	Mega	United	Oryx	Dynamic
Revenue	3	3	0	3	2	3	0	3	0	3	2	3	1	1	3
Profit	1	2	3	3	2	2	0	3	2	3	3	3	3	2	0
Number of new client acquisitions	3	2	2	1	1	2	1	0	2	2	2	0	3	1	2
Extending product offering with new product features/offerings	1	3	3	3	2	1	3	3	2	3	3	2	3	1	2
Number of existing clients presenting for repeat business	3	2	0	1	1	3	3	3	3	2	1	3	3	2	3
Business process management	1	2	1	3	3	1	2	2	2	1	3	2	0	2	3
Customer satisfaction (make the customers more or less satisfied)	2	3	0	2	2	3	3	2	2	3	3	0	3	1	0
Profile of new client acquisitions (larger, smaller, same)	3	2	0	1	2	3	1	0	3	1	0	0	3	1	1
Compatibility with regulatory bodies and local government	0	3	0	1	1	0	0	3	3	3	2	0	0	3	3
Brand image	1	3	2	0	3	2	3	2	0	1	3	3	3	2	2
Changing the nature of customer interaction	0	1	0	2	3	0	0	0	2	0	0	2	0	2	1
Organisational productivity	3	1	1	3	2	1	3	1	3	1	3	0	0	2	2
Employee competence & expertise	3	1	3	3	3	0	3	3	3	3	3	1	2	3	1
Market share in core and allied markets (%age growth or decline)	0	1	1	1	2	2	1	3	0	1	0	0	0	0	3
Pay and perks	3	2	0	3	3	0	0	0	0	0	0	3	0	3	3
Focus on continuous improvement or benchmarking against best practice	3	3	3	3	2	2	3	2	2	1	3	0	0	3	3
Employee Experience	2	1	3	3	3	0	3	3	3	3	3	0	3	3	3
Aid and subsidies, infrastructural support, tax exemptions, etc., from gov.	0	0	0	3	2	3	3	2	0	3	3	0	3	3	3
Skill sets	3	0	3	3	3	1	3	2	0	3	0	0	2	3	3
Organisational Headcount	0	1	0	3	3	0	0	3	1	3	3	3	3	0	3
Retention strategies (especially for precious resources)	3	2	3	3	0	0	0	0	0	3	0	3	3	0	3
Communication effectiveness	0	2	1	2	3	0	0	1	2	3	3	0	3	3	0
Effectiveness of business tools and techniques	3	0	0	3	2	1	3	2	2	0	3	3	0	2	3
Developing or extending internal knowledge repositories	0	3	0	3	3	0	3	2	2	0	0	0	2	0	0
The effectiveness of recruitment and selection procedures	3	3	3	3	1	0	3	2	3	3	3	0	3	2	1
Employee empowerment	3	3	0	1	0	0	3	1	3	3	3	1	2	2	3
The diffusion of knowledge throughout the organisation	0	1	0	2	3	0	3	3	2	0	3	0	2	3	0
Patents, copyrights, branding and trade secrets	0	2	1	1	0	0	0	0	0	0	0	0	0	0	2
Co-operation among people at different levels, functions and departments	0	2	3	3	3	0	1	2	2	2	0	0	2	0	0
Structured and rewarding career paths	3	3	3	0	3	0	0	3	0	3	3	0	0	0	3
Attitude To Work	3	3	3	2	3	0	3	1	3	3	3	0	0	0	3
Training in technical, hard and soft skills, and new technologies	0	0	0	1	3	0	0	0	0	3	3	0	0	0	3
Effectiveness of business documentation	0	3	0	3	0	1	0	1	2	2	0	0	0	1	0
Risk management planning	0	0	3	0	0	0	3	3	0	3	2	1	0	3	0
Career growth	3	0	0	1	3	0	1	0	0	3	3	0	3	0	3
Other specific financial measure	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
Employee qualifications	0	3	0	2	3	0	0	3	0	0	0	0	3	1	0
Active membership in s/w assoc'ns, instit'ns, soc's and communities	3	3	0	0	0	0	3	3	0	0	1	0	0	3	3
Contribution to society (e.g. job opportunities for the less privileged)	0	0	0	0	0	3	0	0	0	0	3	1	3	2	0
Intellectual Agility	0	0	0	0	0	0	0	0	0	0	0	1	0	3	1
Corporate citizenship (self-regulation wrt legal and ethical standards)	0	3	0	0	0	0	0	3	3	3	3	0	0	3	0
Seeking / Retaining a recognised quality standard (e.g. ISO)	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
Rewards / Recognitions	0	2	0	3	0	0	0	0	0	2	0	0	0	0	0
Goodwill, general image among the common public	0	0	0	0	0	3	0	0	2	3	0	0	0	0	0
Registered designs and processes	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0
Research publications	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3
The alignment of training programmes with long term organisational goals	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Training effectiveness	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Change management (i.e. chaperone employees through change events)	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
Assets (not including intellectual capital)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Redressing grievances	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11 Objective achievement per objective per organisation

Modification Value	Modification Interpretation
0	No modification
1	Minor modification
2	Moderate modification
3	Significant modification

Table 12 Modification rating scale for SPI activity and situational change

Survey Insrtument Question	SPI Activity														
	Silverback	Michelin	Grenoble	When	Tribal	BocaJ	Watch	LordHenry	Lakes	Colleran	Cameron	Mega	United	Oryx	Dynamic
1.1.1	2	3	1	3	2	0	1	3	2	2	1	2	1	0	0
1.1.2	0	3	1	3	0	0	0	0	0	1	0	1	0	0	0
1.1.3	0	0	2	0	1	0	0	3	2	0	2	0	2	0	0
1.1.4	1	0	1	0	2	0	0	0	3	0	2	0	1	0	0
1.1.5	0	0	0	0	0	0	0	0	0	3	0	0	0	2	0
1.1.6	0	0	0	2	0	0	1	3	0	2	0	0	0	0	0
1.2.1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1.2.2	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0
1.2.3	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0
1.2.4	0	1	0	0	0	0	0	0	2	0	0	0	2	0	0
1.3.1	0	2	0	3	1	0	0	0	3	3	2	0	0	2	0
1.3.2	0	2	0	0	1	0	0	0	3	3	2	0	0	2	0
1.3.3	1	0	1	0	0	0	0	0	0	2	2	0	0	0	0
1.3.4	0	3	0	0	1	0	0	0	0	0	1	0	2	2	0
1.4.1	0	0	1	2	3	3	2	1	2	2	3	1	0	2	0
1.4.2	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
1.5.1	0	0	0	0	1	0	0	2	0	3	2	0	0	0	1
1.5.2	0	0	2	3	1	0	0	0	0	3	2	0	0	0	0
1.5.3	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0
1.6.1	0	0	2	3	0	1	0	2	0	3	3	0	0	3	0
1.7.1	2	0	1	2	1	0	0	0	0	2	0	0	1	2	1
1.7.2	3	2	1	2	0	0	0	2	2	0	0	0	3	2	3
1.7.3	0	0	1	2	1	0	0	2	2	1	0	0	0	0	0
1.7.4	2	3	0	1	0	0	0	1	2	2	1	0	0	0	0
1.7.5	1	0	0	3	0	2	1	3	1	2	3	0	0	2	0
1.7.6	0	0	0	3	1	2	0	0	0	3	0	0	0	0	0
2.1.1	0	3	0	0	0	0	0	2	2	0	1	2	1	0	0
2.2.1	0	3	2	0	1	1	0	0	3	2	2	1	1	0	2
2.3.1	0	0	2	0	2	0	0	2	1	0	1	1	3	0	0
2.4.1	1	0	0	3	2	0	0	0	0	2	1	0	2	0	0
2.4.2	2	2	1	2	0	0	0	2	0	2	3	0	3	0	0
2.4.3	0	0	1	0	2	0	0	1	2	3	0	0	0	0	2
2.5.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.1.1	2	0	0	3	0	3	2	0	3	2	2	0	0	0	2
3.1.2	0	0	0	1	0	0	0	0	2	0	1	0	0	0	0
3.1.3	2	3	0	3	0	0	2	2	1	0	3	0	3	0	3
3.2.1	0	0	1	3	0	1	2	1	0	0	2	0	3	0	0
3.3.1	2	1	2	3	0	0	0	2	0	0	3	0	0	0	0
3.4.1	1	0	0	0	0	1	0	0	2	0	0	0	0	0	0
3.5.1	0	0	0	2	1	0	0	0	2	3	2	0	0	0	0
3.6.1	0	0	0	0	0	0	3	0	0	0	0	0	0	1	0
3.7.1	0	0	0	0	0	0	3	0	0	2	0	0	0	0	0
4.1.1	1	0	0	0	2	2	1	0	2	3	2	0	0	0	2
4.2.1	0	0	0	0	0	0	1	0	0	1	3	0	0	3	2
5.1.1	0	0	1	3	1	0	0	2	3	1	3	0	0	0	0
5.1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.2.1	2	0	1	3	0	0	0	2	2	1	2	0	0	0	3
5.3.1	1	3	1	1	1	2	1	2	0	3	2	0	0	2	0
5.4.1	0	0	1	1	0	0	0	0	0	2	3	0	0	0	0
5.4.2	0	0	1	1	0	0	0	2	0	0	3	0	0	0	0
5.4.3	0	0	0	1	1	1	0	2	2	0	2	0	0	0	0
5.5.1	2	2	0	3	0	0	0	1	0	0	2	0	0	0	0
5.5.2	2	0	0	3	0	0	0	0	0	0	0	0	0	0	0
6.1.1	2	0	1	0	0	0	0	2	3	1	3	0	0	1	3
6.1.2	1	3	1	0	1	0	0	1	0	2	1	0	1	0	3
6.2.1	0	3	0	2	0	0	0	2	2	3	2	0	1	0	3
6.2.2	0	0	0	1	0	0	0	0	0	0	1	0	0	0	3
6.3.1	2	0	0	0	0	0	0	0	2	1	0	0	0	0	0
6.4.1	0	0	1	1	1	0	2	0	0	0	1	0	0	0	2
6.5.1	1	0	0	3	0	0	0	3	2	0	3	0	0	0	0
6.6.1	0	0	0	3	0	0	0	2	1	2	0	0	0	0	1
7.1.1	2	0	0	0	0	0	0	2	0	0	1	0	0	0	0
7.2.1	3	0	1	3	2	2	0	3	1	0	3	0	1	0	3

Table 13 SPI activity as recorded on modification scale

	Situational Change														
Survey Instrument Question	Silverback	Michelin	Grenoble	When	Tribal	Bocal	Watch	LordHenry	Lakes	Colleran	Cameron	Mega	United	Oryx	Dynamic
1.1.1	3	2	2	1	2	2	0	0	3	0	2	2	3	2	2
1.1.2	1	1	0	0	1	0	3	1	0	0	3	2	3	0	0
1.1.3	2	0	0	2	3	0	2	0	2	3	3	1	2	0	3
1.1.4	3	2	0	3	2	0	1	1	3	1	2	0	2	1	1
1.1.5	2	2	0	0	1	1	2	3	1	3	1	0	3	2	0
1.1.6	0	1	0	3	2	0	1	1	2	3	3	1	0	2	0
1.1.7	0	1	0	3	1	0	1	0	2	1	0	1	0	0	2
1.1.8	0	2	0	0	2	0	2	0	1	1	0	1	2	0	0
2.1.1	0	2	2	0	3	3	1	3	0	3	3	1	0	3	3
2.1.2	0	1	2	1	0	0	1	0	0	0	0	1	1	2	0
2.1.3	0	2	0	0	1	0	0	1	0	0	1	1	0	0	0
2.1.4	2	1	1	1	2	0	0	0	1	2	2	1	1	1	0
2.1.5	0	1	2	0	1	2	0	1	1	2	0	1	0	1	0
2.1.6	1	1	2	1	0	1	2	0	1	1	2	1	0	0	3
2.1.7	0	0	1	0	0	1	0	0	0	2	0	0	0	3	0
2.1.8	0	1	0	0	0	0	0	0	3	0	1	0	1	1	0
2.1.9	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0
3.1.1	0	3	0	0	0	0	1	2	2	2	1	0	2	0	0
3.1.2	3	2	0	0	2	1	0	0	1	0	1	0	2	0	0
3.1.3	0	3	0	2	0	0	0	1	1	2	0	2	0	0	0
3.1.4	0	3	0	2	3	0	0	0	0	2	0	0	0	0	0
4.1.1	3	1	2	2	2	3	1	0	0	3	2	0	0	3	3
4.1.2	3	0	0	1	3	3	1	2	0	0	3	0	0	1	3
4.1.3	3	2	2	3	3	3	1	3	0	3	2	1	0	3	3
4.1.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4.1.5	2	1	0	2	0	1	0	0	0	0	2	0	0	0	0
4.1.6	0	0	3	3	2	1	1	2	0	2	3	0	0	1	1
5.1.1	0	1	3	0	2	0	0	0	2	0	0	1	0	0	0
5.1.2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5.1.3	3	2	1	0	2	2	3	3	0	3	3	2	2	2	3
5.1.4	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
5.1.5	1	0	0	0	1	1	0	2	0	0	2	0	0	0	0
5.1.6	1	1	0	1	2	0	0	0	0	0	3	0	2	0	3
5.1.7	0	2	2	0	2	1	0	2	2	2	3	1	2	2	3
6.1.1	0	0	1	3	0	0	1	2	0	3	2	1	0	0	0
6.1.2	1	0	0	2	1	0	0	2	0	3	2	2	1	0	0
6.1.3	1	0	0	0	0	0	0	3	0	0	3	2	0	0	0
6.1.4	0	1	0	0	2	0	0	0	2	0	3	0	0	0	0
6.1.5	0	2	0	0	0	0	0	0	0	1	3	1	2	0	0
6.1.6	0	2	0	1	0	0	0	1	2	0	2	1	3	1	2
6.1.7	0	0	2	0	0	0	2	1	3	1	2	1	3	1	2
6.1.8	2	0	2	1	0	0	2	0	2	0	2	1	0	1	0
6.1.9	0	0	0	1	0	0	0	0	0	0	2	1	1	1	1
7.1.1	3	0	3	2	2	0	3	1	3	1	2	1	0	0	2
7.1.2	3	1	3	3	1	0	2	0	3	3	3	1	2	1	1
8.1.1	2	0	0	0	0	1	0	2	1	2	3	0	0	3	1
8.1.2	2	0	3	3	0	0	0	2	0	0	3	0	0	3	0
8.1.3	2	0	2	0	0	0	0	0	1	0	2	0	0	1	0
8.1.4	2	0	2	1	1	1	0	2	1	0	3	0	0	3	0

Table 14 Situational change as recorded on modification scale

7.8 Summary

This chapter has outlined how the data collection was conducted for the study. Initially, the challenge of accessing organisations was discussed, followed by some details of how participant confidentiality requirements were addressed. The profile of the participating organisations and individuals was also outlined, along with information regarding the data collection periods and durations. The latter part of the chapter presented a summary of the actual data collected.

This chapter is the final chapter in Part 3 of the thesis, which is dedicated to describing the data requirements for the study as well as the mechanism for collecting and rating the data from the participating organisations. In Part 4, the data analysis and evaluation details are provided, with a focus on study findings.

Part 4 Research Findings

The fourth part of the thesis contains two chapters. Chapter 8 presents the details of the analyses conducted on the study data. In chapter 9, the results of the data analyses are evaluated for meaning, focusing particularly on the association between SPI activity, situational change and business success.

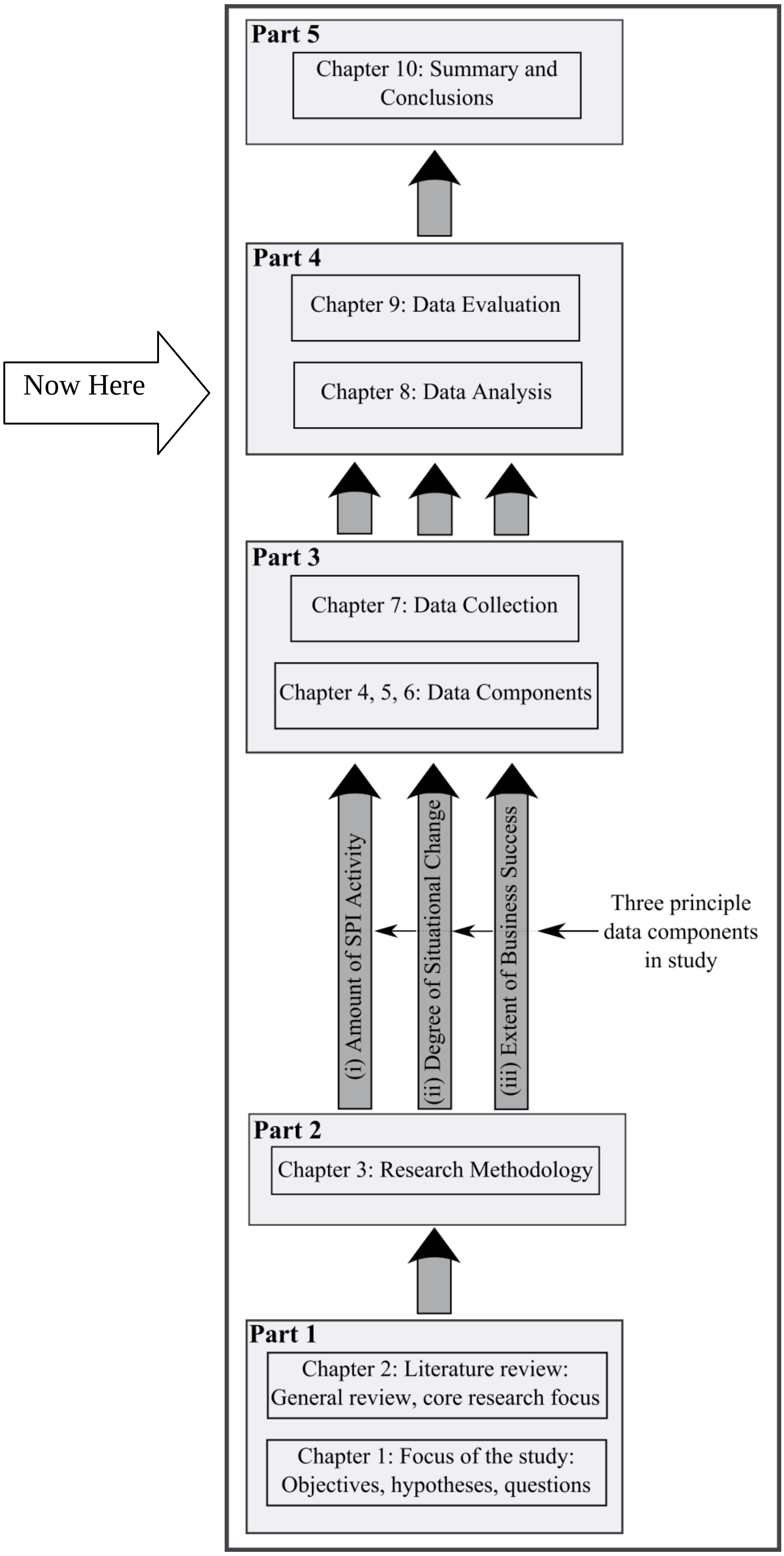


Figure 21 Map of Thesis – Part 4

8 Data Analysis

8.1 Introduction

In this chapter, the approach to quantifying the three principle phenomena under investigation is outlined: the extent of business success, the amount of SPI activity, and the degree of situational change. Thereafter, an analysis of the data examines the most and least common areas for SPI activity and situational change, along with an analysis of the business objectives for the participating organisations. The latter part of this chapter examines the co-relationship between SPI activity and business success, and finally, the influence of situational change is analysed.

8.2 Quantifying business success

In chapter 6, the two phased approach to examining business success was outlined. During the first phase, the business objectives for the forthcoming year were determined. The second phase was discharged at the end of the year under investigation, determining the extent of achievement of the previously recorded objectives. Chapter 7 provided details of how a survey instrument was utilised to collect the business objectives and (later) their achievement, with Table 9 identifying the objectives and agreed ratings for each of the participating organisations. Furthermore, in Table 11, the extent of achievement for each of the recorded objectives is presented. This section provides and explanation of how an overall measure of business success was calculated using the underlying data in Table 9 and Table 11.

When making determinations in relation to the extent of business success, it is necessary to consider two pieces of information: (1) the strength of the original objectives; and (2) the extent of achievement of the objectives. In order maximise the validity of the data analysis, this research established two separate approaches to measuring business success. Firstly, business success is quantified in terms of the extent of achievement of each of the individual objectives. Under this intuitive *basic* interpretation of business success, the overall measure of success for an organisation is increased each time one of their objectives is achieved to any extent. Furthermore, the achievement of higher rated objectives results in relatively larger increases in the overall success measure. For example, the total achievement of a high priority objective with an explicit target will increase the overall success measure by 4 (4 [value for a high priority objective with an explicit target] x (3 [actual achievement value] / 3 [total achievement value])). In contrast, the total achievement of a low priority objective with no explicit target will increase the overall success measure by 1 (1 [value for a low priority objective with no explicit target] x (3 [actual achievement value] / 3 [total achievement value])). The summing formula for quantifying the overall basic success score for an organisation is as follows:

$$BasicBusinessSuccess = \sum_{i=1}^{N} ObjectiveRating(i) * \frac{ObjectiveAchievement(i)}{TotalAchievementValue}$$

Where i is an individual business objective and N is the total number of business objective questions (as outlined in Chapter 6, there are 51 business objectives questions). Table 9 presented the collected data regarding the objective ratings for the participating organisations, while Table 11 presented the extent of achievement for recorded objectives. The $TotalAchievementValue$ is the value for the maximum possible achievement for an objective (as outlined in Table 10, the totally achieved value is 3).

While the basic business success interpretation is useful in measuring the success of an individual organisation, it has a limitation when examining the relative business success across multiple organisations. Since the basic business success interpretation does not have a mechanism for lowering the overall score for instances of failure to achieve objectives, it is possible for an organisation with a large number of stated objectives (some of which are not achieved) to appear more successful than another organisation with fewer objectives (all of which are totally achieved). Clearly, this could be problematic as some organisations may list relatively large numbers of objectives while others may list relatively few. In order to address this issue and to increase the overall analytical strength of the research, an additional approach to quantifying business success was introduced.

The second approach to quantifying business success involves the introduction of a *weighted negative marking* (WNM) scheme. Under this second interpretation, an organisation still receives a weighted score for each objective that is achieved to some extent; however, unlike the basic interpretation, an organisation receives a weighted negative score for each objective that is not achieved or only partially achieved. This type of WNM acts as a counter-balance; in cases where there are large numbers of objectives which are not achieved (or only partially achieved), the overall score will reflect the extent to which stated objectives were not achieved. The essential benefit, therefore, of the WNM interpretation is that it facilitates the punitive measure of lowering overall success scores if there are objectives that are not achieved or only partially achieved. Furthermore, the weighting scheme is designed such that individual weights are in proportion to the strength of the objectives. Applying the WNM scheme outlined in Table 15, the non-achievement of a low priority objective with no specific target will result in a -1 being applied to the overall WMN business success score, whereas the non-achievement of a high priority objective with an explicit target will result in a -4 being applied to the overall WMN business success score.

			Objective Rating				
			No Objective	Low Priority	Low Priority Explicit Objective	High Priority	High Priority Explicit Objective
			0	1	2	3	4
Achievement	Not achieved	0	0	-1	-2	-3	-4
	Partially achieved	1	0	-0.67	-1.33	-2	-2.67
	Mostly achieved	2	0	0.67	1.33	2	2.67
	Totally achieved	3	0	1	2	3	4

Table 15 WNM business success scoring scheme

With the participating organisations reporting varying numbers of objectives in the first instance, WNM ensures that a company with many objectives cannot score relatively highly unless they are achieving well in most or all objectives. By quantifying business success using WNM, the resultant business success score can potentially provide a fairer and more accurate representation of the relative business success of the participating companies – and it is the dimension of relative success that is of primary importance to this study. However, the basic business success score is merited in that it generates a non-adjusted score for the achievement of stated objectives, and therefore, it is also considered to provide a valid measure of business success. Since there are benefits to both interpretations of business success, both are retained in the study. This facilitates two separate views of business success, hence improving the reliability of the findings when conducting data analysis and evaluation.

Using the WNM scheme presented in Table 15, the WNM business success score is quantified using the formula:

$$WNMBusinessSuccess = \sum_{i=1}^{N} WNM(i)$$

As with the basic business success score, i is an individual objective and N is the total number of business objectives questions.

8.3 Quantifying SPI activity

In addition to quantifying business success, the amount of SPI activity and the degree of situational change must also be determined. As presented in chapter 4, SPI activity is examined using an ISO/IEC 12207-based survey instrument. This survey instrument was deployed to the participating software SMEs, with the data collected presented in Table 13. Participants quantified individual modifications to the software process according to the Likert scale presented in Table 12. In order to quantify the overall amount of SPI activity in an organisation, the individual recorded instances of process modification are summated into a total SPI activity measure, as follows:

$$SPIActivity = \sum_{j=1}^{M} ModificationValue(j)$$

Where j is an SPI activity and M is the total number SPI activity questions in the survey instrument (as outlined in Chapter 4, there are 63 of these SPI questions).

As discussed in chapter 4, SPI activity is concerned with determining how much overall software process change has occurred in a setting over a period of time. Earlier approaches to SPI tend to have focussed on process maturity and not on SPI activity. Therefore, the creation of the *SPIActivity* summation (identified above) represents a new type of measurement for software development.

8.4 Quantifying situational change

In the case of situational change, the broad range of situational factors affecting the software development process identified in chapter 5 are examined, recording instances of reported changes to situational factors. The situational change data collected from the participating companies is presented in Table 14. Where there is a reported situational change, the interviewee indicated the extent of the modification to the situational factor according to the Likert scale presented in Table 12. When quantifying the total amount of situational change in an organisation, the following summation is adopted:

$$SituationalChange = \sum_{k=1}^{L} ModificationValue(k)$$

Where k is a situational change, and L is the total number of situational change questions in the survey instrument (as outlined in Chapter 5, there are 49 of these situational change questions).

The measurement of the amount of situational change in a software development setting is a new concept introduced by this research. Therefore, the *SituationalChange* summation identified above is another new type of measurement for software development.

8.5 Tools used in the data analysis

A number of different tools were adopted to support the data analysis effort. Initially, and as highlighted in chapter 7, Microsoft Excel was used to consolidate all the data collected from the field. Within MS Excel, it is also possible to conduct some basic data analyses, including the generation of totals and averages, and the visualisation of data sets. However, in order to conduct more detailed statistical analyses, it was necessary to use specialised statistical analysis tools, including *STATISTICA*[5] and *R*[6]. Using such specialised data analysis tools, it is easier to assess the distribution of the data, and to perform various statistical correlations and probability calculations.

[5] STATISTICA is a trademark of StatSoft Incorporated.
[6] *R* is a programming language and a software environment for statistical computing, freely available under the GNU General Public Licence.

The following section presents an overview of the data, followed by the basic data analysis that was conducted using MS Excel. Thereafter, details of the more specialised statistical analyses are presented, including the results of the statistical correlations.

8.6 Overview of data

For each of the data components, data quantification is conducted as per the summation functions outlined above. In general, for each of the three phenomena under investigation, a considerable degree of variation is evident within the participating organisations (refer to Figure 22, Figure 23, Figure 24 and Figure 25).

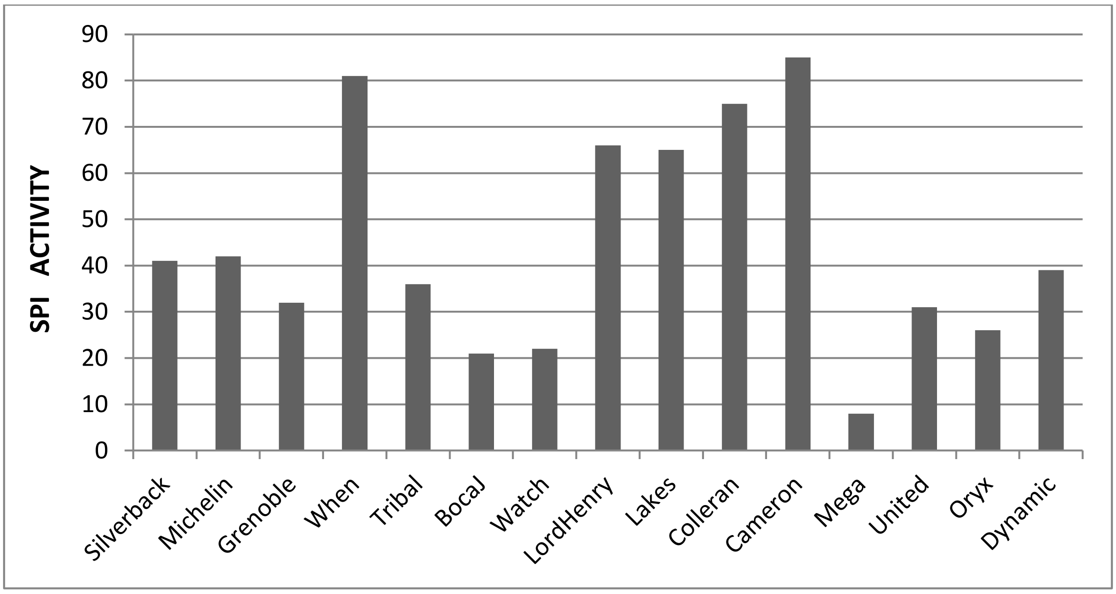


Figure 22 Amount of SPI activity

As can be seen in Figure 22, there is a large variation in the reported amount of SPI activity. For example, the lowest amount of recorded SPI activity is in the organisation *Mega*, which has an SPI activity value of just 8. However, two of the organisations report over ten times this amount of SPI activity: *When* and *Cameron*. This data informs us that there is a considerable difference in the amount of SPI activity in the participating organisations, which in itself is an interesting discovery.

In Figure 23, it can be seen that there is also quite a large degree of variation in the situational change reported in the participating organisations. For example, the organisation *BocaJ* has the lowest reported degree of situational change with a value of 28. However, with a situational change value of 87, the organisation *Cameron* has over three times the amount of situational change recorded in *Boca*. The data therefore informs us that there is a large degree of variation in the extent of situational change in the participating organisations - although there is less variation than in the case of SPI activity (as can be seen by visually comparing Figure 22 and Figure 23).

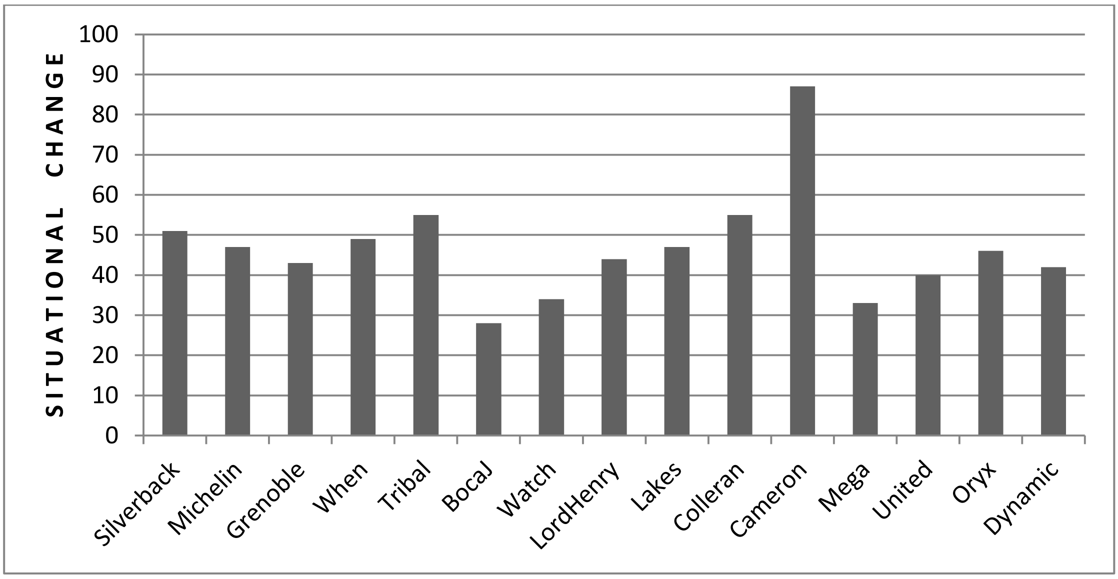


Figure 23 Degree of situational change

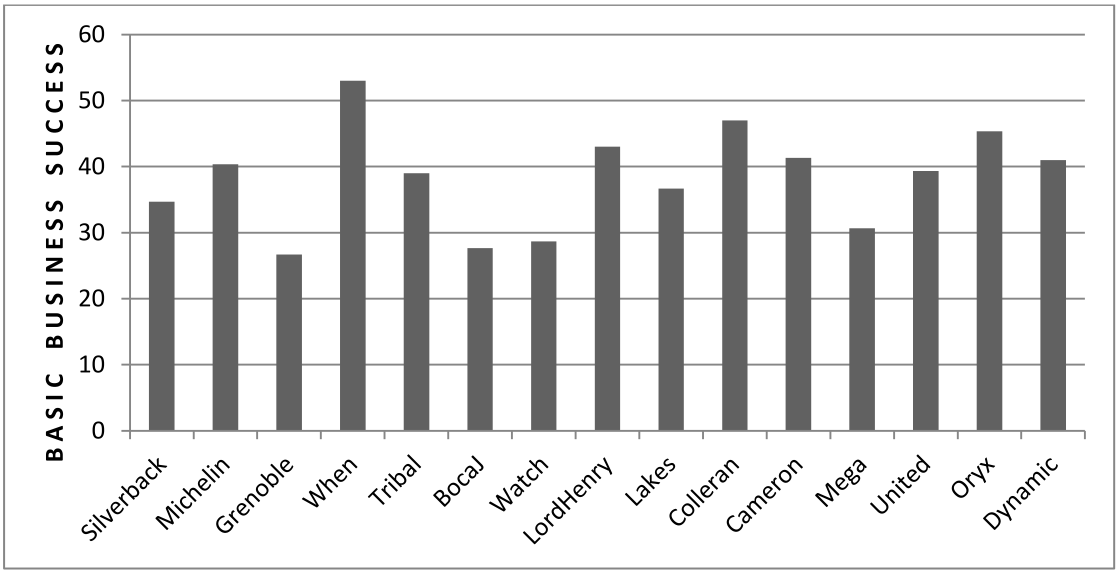


Figure 24 Extent of business success – Basic

Figure 24 illustrates that there is also variation in the extent of business success that is reported. Under the basic business success interpretation, this variation is not especially large. The lowest recorded basic business success value is for the organisation *Grenoble*, which has a basic business success value of 26.67. The largest recorded basic business success value is 53 in the organisation *When*. However, under the WNM business success scheme, the variation is much more pronounced (refer to Figure 25). The lowest WNM business success organisation value is 2.67, again for the organisation *Grenoble*. However, five of the participating organisations have a WNM business success value that is over ten times the WNM business success of *Grenoble*, namely *When*, *Tribal*, *LordHenry, Lakes* and *Cameron*.

This variation demonstrates the power of the WNM scheme to adjust the relative success measure in cases where organisations had failed to achieve objectives. For example, the organisation *Oryx* had a basic business success value of 45.33 (the third highest) but only records a WNM business success value of 17.33 (the fourth lowest). In the case of *Oryx*, many business objectives were recorded in the initial business objectives interview. While some of these objectives were achieved, many other objectives were not and as a result the WNM business success value for *Oryx* has adjusted the relative success downwards.

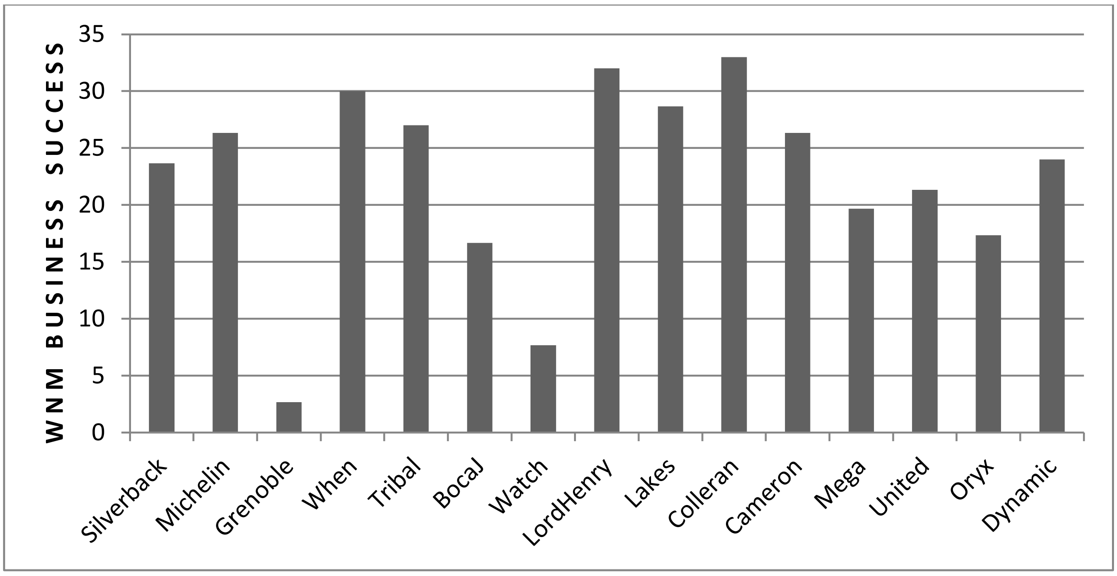


Figure 25 Extent of business success – WNM

While the above bar charts demonstrate some basic properties of the data, such as highest and lowest values, and the degree of variation, it is necessary to conduct more advanced data analyses in order to better understand the nature of the relationship between these different phenomena.

8.7 Basic analysis of data

The basic analysis of the data that is presented in this section focuses on profiling SPI activity, situational change and business success across the participating organisations. The following section will present details of the statistical correlations, with the following chapter (chapter 9) dedicated to evaluating the impact of the data analysis findings.

Across the fifteen participating organisations, different SPI activities and situational changes are reported. Furthermore, different business objectives are also evident in the various companies. A basic analysis of the study data provides some interesting insights into the type of SPI, situational change and business objectives that are evident across the participating companies.

8.7.1 SPI Profile

While a broad spectrum of SPI was reported in the study, certain process areas tend to receive greater attention. Following an analysis of the reported SPI activity, a hierarchy of SPI for software SMEs was developed (refer to Figure 26).[7] The following subsections present some details on the most common and least common processes undergoing SPI in the participating companies.

8.7.1.1 Most common areas for SPI activity

As well as being reported as the most common process area for SPI in the participating SMEs, the requirements analysis process is also generally reported as undergoing the most significant amount of process change. For the most part, the reported improvements relate to a movement towards a more stringent requirements specification process. As a result, it was reported that there has been an emphasis on improving *"the quality of the definition of the requirements"*, and to including *"more*

[7] The hierarchy of SPI pyramid was constructed by analysing the responses from all participants to each question in the SPI survey instrument. Both the frequency and the amount of reported SPI in a specific process are jointly considered, with processes experiencing more frequent and/or more significant SPI being placed higher on the pyramid. The tiering of the pyramid was a natural outcome of the clustering that was evident in the data, i.e. the analysis demonstrated that there were points in the data where relatively large gaps were observable in the frequency and/or significance of the reported SPI.

detail… on what's included and what's excluded".[8] Furthermore, companies have recognised that in the past, proceeding to implementation prior to requirements sign-off was problematic, with one particular organisation identifying this as previously being *"quite a big issue"*. Therefore, SPI has focused on getting requirements *"sign[ed] off before we've gone and built stuff"* or as another organisation put it, *"before any solid code has been developed"*.

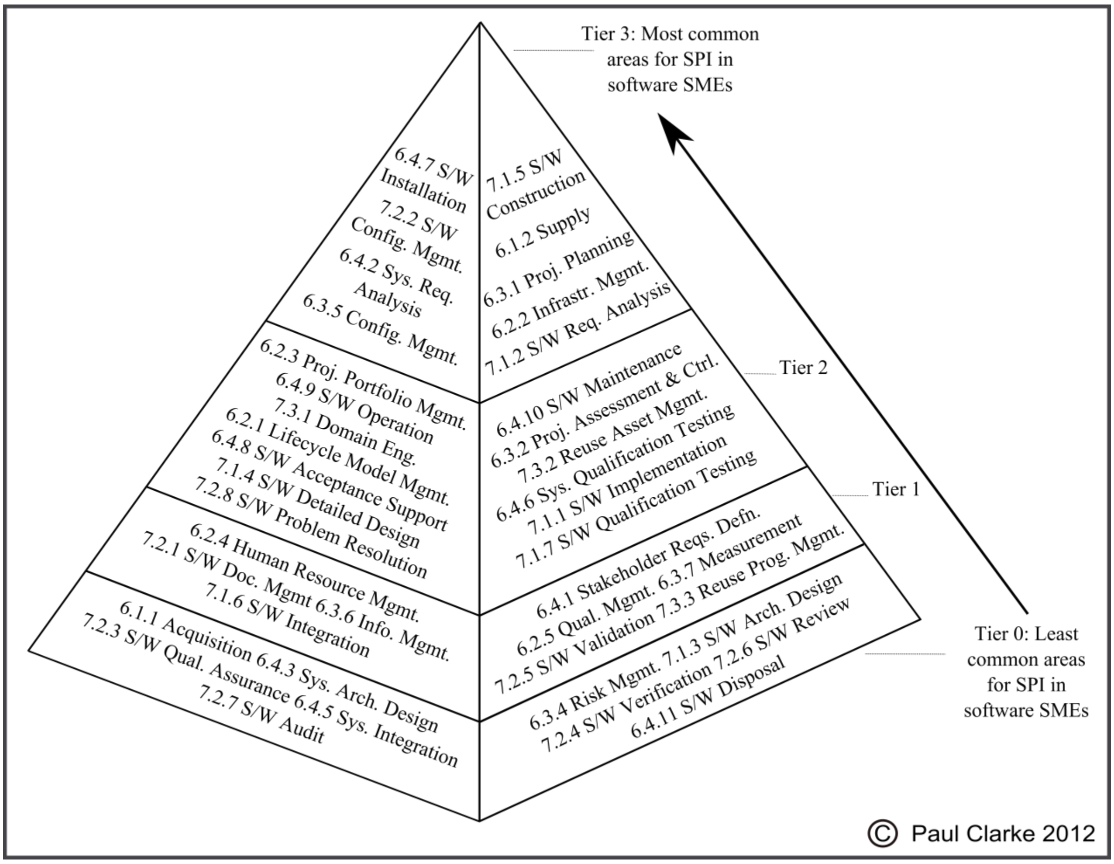


Figure 26 Hierarchy of SPI for software SMEs

The provision of an infrastructure for development and for operations is also another major area for process change in the study. In this regard, a major consideration relates to the decision to outsource or off-site some (or all) of the hardware infrastructure, with one of the organisations reporting that *"it's all offsite now, it's just easier"*. However, a different company reports that they have *"taken that over ourselves"* (having previously used a partner for hardware infrastructure). With four of the participating organisations reporting changes to the utilisation of third parties

[8] Note that this section makes extensive use of the commentary that was collected throughout the interviewing process. Quotation marks indicate a comment from a study participant.

for infrastructure provisions, it is clear that this is an important area for process improvement in software SMEs. Furthermore, implementing a change of infrastructure provisioning arrangements can be a non-trivial event, with one of the companies reporting that the exercise was *"pretty painful"*. Predominately, there would appear to be a movement towards using third parties for the provision of infrastructure requirements, a phenomenon that is perhaps related to an increase in the number of data centres and web-based applications. A number of other organisations also report minor or moderate changes in the approach to infrastructure provision, with one company transitioning all staff to a common development environment.

The approach to tendering, bidding and negotiating with clients is also reported as undergoing a considerable amount of improvement. For a number of the participating SMEs, this is related to the emergence of partnering arrangements, whereby multiple companies will partner in order to enhance their offering to potential clients. In this respect, one company reports that they are presenting themselves to the market as *"more of a solutions company"*, while another participating organisation reports that considering new reseller agreements, the sales process now includes a *"decision… as to whether we bring [a third party] into the sale"*. There is also a confirmation from two other participating organisations that they have improved their tendering and negotiating, stating that there is *"a little more conversation… over milestones and payments… as well as the detail"*, and reporting that they are *"more stringent about making sure that we're only going after profitable deals, or ones that are a strategic move – we're just trying to be more cautious"*. Another organisation reports that they have developed a *"different offer"* altogether for the market.

A number of the participating organisations also report minor or moderate changes in the approach to software installing, and planning for installation, in the target environment. In the case of two organisations, they report increases in the degree of automation associated with installations. Another organisation has highlighted an increased need for frequent releases, stating that over the past year, there have been *"a lot more releases in stages"* and that they are now *"more structured for those types of projects"*. A further organisation highlights that there was a *"change in the installation strategy… due to downsizing [of the company itself]"*.

In addition to the common areas identified above for SPI activity, the participating organisations also report that there is an increased consideration for the feasibility of operations and maintenance. In order to reduce future support and maintenance costs, a number of organisations report that they have increased levels of refactoring. Another organisation reports that they *"are looking to cost more accurately the ongoing maintenance and support costs"*, with a further company again stating that in order *"to reduce the maintenance burden... we're much more conscious, from the requirements point of view, of the support burden"*. As is discussed in more detail in chapter 9, there is a difference between classical SPI in terms of actual improvement actions that can be observed in the software process and those improvements that are more of a tacit nature. While tacit improvement may not be immediately measurable in the process as an observable effect, improvements in the mental check-listing of process participants does represent an improvement to the method of work.

Although one of the organisations reports that owing to decreases in company headcount, there was a reduction in testing activity, the majority of the organisations report general improvements in testing. For some organisations, this is simply that the *"test suite is growing slowly over time"*. For other organisations this is a more explicit improvement to the test process, with one respondent stating that *"testing is more well defined and more rigorous than it had been"* and other respondent confirming that *"we're allocating more time at the development stage to performance testing than we would have done a year ago"*. Another organisation reports that they are *"measuring more the code coverage of our automated tests"*. Some of the participating companies also report improvements in the effort estimation process, stating that the estimates are *"more well defined than they would have been"* and that there is more team involvement in estimation exercises. Greater team involvement in estimation is reported by one respondent to *"give ownership [for estimates] back"* to the team, with the result that estimates are becoming *"more accurate"*.

The participating organisations also report improvements in the approach to performing change requests (CRs). Describing the company as previously having been *"very ad hoc"* in the handling of CRs, one organisation has enforced the recording of CRs in a tracking system. Another organisation reports that they are *"a lot more rigorous"* about change management for requirements, with related increases in the visibility and tracking of CRs. Some of the improvements in visibility and tracking of CRs have been realised through the adoption of new tooling support. As is discussed in more detail in the chapter 9, improvements in tooling are often reported by the participating SMEs as enablers of improvements in the approach to the full life cycle of activities associated with software development.

8.7.1.2 Least common areas for SPI activity

Not one of the participating organisations reported improvements in the approach to collecting and maintaining quality cost data, with one organisation providing the insight that they *"haven't a clue"* regarding the cost of quality. While other organisations weren't quite so explicit in relation to the approach to costing quality, the indifference that was evident in the responses suggests that it may be an area which receives very little attention in software SMEs. Similarly, the accumulated evidence of this research suggests that software SMEs pay little attention to the disposal of software. The participating organisations reported that they do not have to be concerned with the secure disposal of software or with decommissioning software systems. Where sensitive data was contained in systems, the organisations reported that the system itself automatically managed the data or that the client was responsible for such concerns.

In general, participants reported a strong tendency to merge architectural and design considerations. Predominately, they report that from their point of view, the design and the architecture were not treated as separate concerns, but rather that the design was the architecture and vice-versa. The software SMEs in the study also report little change to the process of reviewing process quality procedures or to reviewing process implementation in general. Again, the indifference evident in the responses suggests that these are areas to which the participating SMEs pay little attention. This suggests that the software development process evolves in a non-structured fashion rather than in a focused or highly controlled manner in software SMEs.

The analysis of the study data also reveals that that participating organisations have made almost no improvement to the process of independently auditing either projects or software deliverables. Quite a large number of the participating organisations asserted that, in fact, they never perform independent audits of their projects or products. The participating companies also report very low levels of process improvement with respect to the development of designs for external interfaces and databases. There was no evidence of any major initiatives to improve the approach to defining data and databases over the study period, and no reported improvement in the definition of preliminary test requirements at the requirement capture stage.

8.7.2 Situational change profile

A basic analysis of the study data also reveals that some aspects of the situational context are routinely reporting relatively large degrees of change while other aspects of the situation are subject to only minor change (or in some cases, no change at all). This section presents details of the most and least common areas for situational change. An overview of the hierarchy of situational change is presented in Figure 27, with the following subsections elaborating on the most and least commonly reported situational changes in this research.[9]

8.7.2.1 Most common areas for situational change

Staff headcount is the single greatest area for reported situational change in the study. All of the participating organisations reported changes to their headcount and predominately this relates to increases, with 11 of the participating organisations reporting additions to headcount levels over the study timeframe. In some cases the increases were modest, but 9 of the participating companies reported headcount increases of 25% and greater over the year under investigation. Two of the organisations that experienced headcount reductions lost 40% or more of their staff over the year. These reported changes to headcount figures represent significant fluctuations, highlighting the volatile nature of business in software SMEs. Such volatility is a major catalyst for process change, and may even suggest that software

[9] The hierarchy of situational change pyramid was constructed by analysing the responses from all participants to each question in the situational change survey instrument. Both the frequency and the amount of reported change in each situational aspect are jointly considered, with more frequent and/or more significant situational changes being placed higher on the pyramid. The tiering of the pyramid was a natural outcome of the clustering that was evident in the data, i.e. the analysis demonstrated that there were points in the data where relatively large gaps were observable in the frequency and/or significance of the reported situational change.

SMEs are more in need of active process management than larger, more stable organisations.

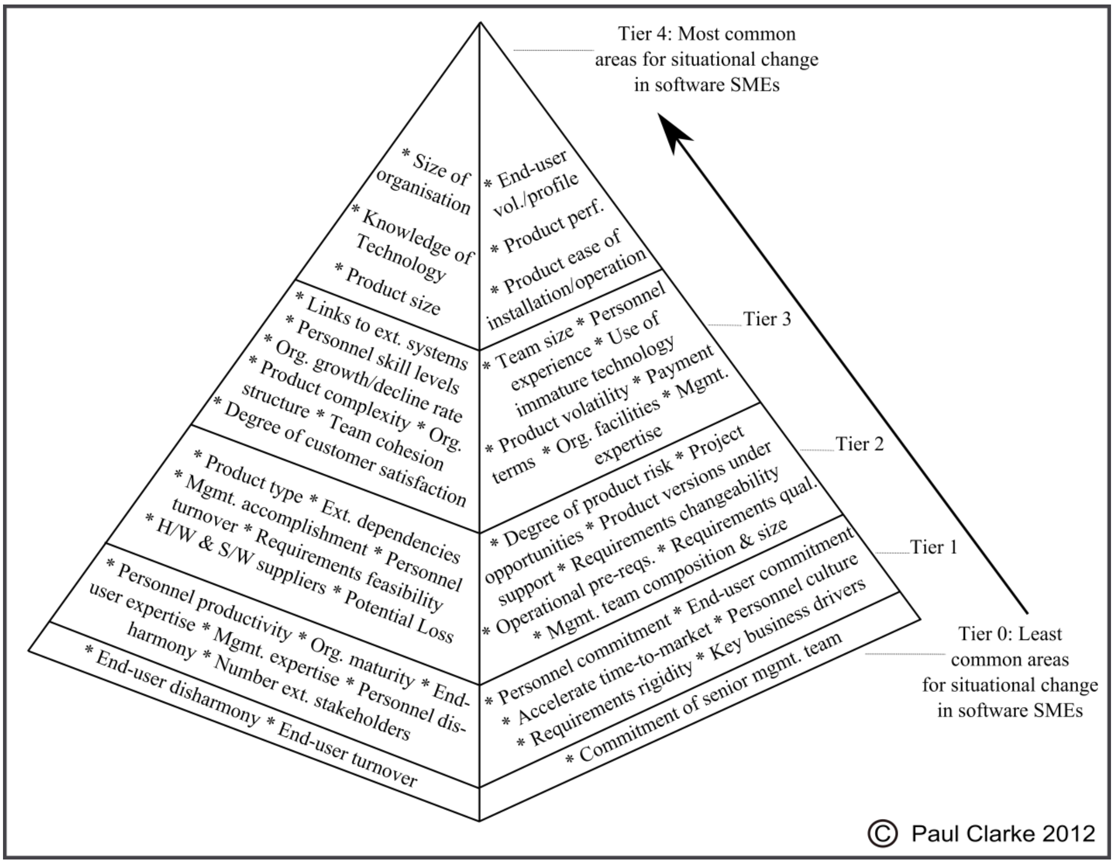


Figure 27 Hierarchy of situational change for software SMEs

The participating organisations also report considerable change in the volume and profile of end users of their software products. Increases in the volume of end users or in the net volume of transactions processed are reported in 11 of the participating organisations, and in some of these cases the organisations report a significant increase in the number of end users or volume of transactions that their products must support. Interestingly, none of the organisations report a reduction in the number of end users or volume of transactions in their products. Furthermore, two of the participating organisations report changes to the profile of end users, that their products now need to cater for different types of end users.

Of the participating organisations, 13 companies reported increases in the knowledge of technology. In some cases, this involved supporting new operating systems, such as Linux and emerging mobile device operating systems. In other cases, these changes were more focused on the software development infrastructure, including changes to integrated development environments and changes of compilers. While the extent of the reported change in knowledge of technology varies across the participating organisations, almost all of organisations are increasing their knowledge of technology mostly through the adoption and utilisation of emerging technologies.

Most of the participating organisations also reported an increase in the required performance of their products, with 12 of the companies reporting varying degrees of increase in the required product performance. Those organisations that do not report increases in performance requirements do not report decreases either, but rather that the performance requirements remain unchanged. In terms of the size of the products, 10 of the participating organisations report increases in one form or another. For some organisations, this increase takes the form of increased data storage requirements while for other companies the reported change relates to the size of the code base. Of interest, one of the organisations reported a slight decrease in the code base as a result of an intensified refactoring effort. A total of nine of the participating organisations reported that there is an increase in the required ease of installation and operation of their software products – hence the need to continually improve the installation procedures and to constantly try to maximise the end user experience of the product(s).

8.7.2.2 Least common areas for situational change

Of all the situational factors examined in the study, just one factor was reported as being completely unchanged in all of the participating organisations: the commitment of the senior management team to projects. In this respect, it should be noted that predominately, the personnel participating in the study were senior managers. Therefore, it is not likely that they would divulge a lower commitment to an external party or indeed, that they would report lowering levels of commitment across the management team. Equally, there is a possibility that some of the participants were biased in their reporting of their own levels of commitment. This may be especially the case for the CEO or owner of the organisation whose remit involves sustaining maximum commitment levels among the senior management team.

Two of the participating organisations report that they increased or decreased the number of external stakeholders over the period of investigation. In one case, this was the result of now working with systems integrators on a more regular basis. In the case of the second organisation, the opposite effect was reported: systems integrators were no longer being used to deploy systems but rather the company had started to engage directly with its end-clients. However, 13 of the participating companies reported no change with respect to the number of external stakeholders, and therefore, in general, it would appear that this is not a major area for change in software SMEs.

There was also little reported change regarding the turn over of end-users of the products. For some organisations, this was owing to the nature of their product. For example, several organisations deployed solutions into a large network of software systems such that the solution itself had little or no direct end-user interaction. This was the case for a number of the telecommunications participants whose software systems were not directly accessible by end-users – since the end users actually used mobile devices to access services in the telecommunications network. In another case, a data mediation platform was the core product from a participating company – and this was not a product that was generally accessible by a large number of end-users, but rather just a few specialised users would configure and administer the product.

The participating organisations also reported little or no change in the personnel culture, with the personnel disharmony levels (including interpersonal conflicts) remaining largely the same as in early periods.

8.7.3 Business success profile

In Section 8.6, the data analysis demonstrated that there is a considerable degree of variation in the amount of business success evidenced in the participating organisations, especially under the WNM business success interpretation. As outlined in chapter 6, the extent of business success is measured using a two-phased engagement that is under-pinned by the HSC. The first phase was dedicated to collecting the business objectives for the participating organisations with the second phase, 12 months later, focussing on determining the extent to which the stated objectives were achieved. An analysis of business objectives from the initial phase demonstrates that some business objectives would appear to be more important, while other objectives are consistently recording a low priority (refer to Figure 28).[10]

8.7.3.1 High priority business objectives

The participating SMEs consistently have high priority objectives in six key areas: *revenue*, *profit*, *extension of product offerings*, *new client acquisitions*, *repeat business from existing clients*, and *business process management*. Growth in revenue is the single most important objective for SMEs, followed closely by profit considerations. Objectives in relation to profitability appeared to be somewhat eclipsed by a more basic need for survival – highlighting the difficult operating realities faced by some software SMEs. After revenue and profit targets, the next highest priority objectives are reported to be the extension of product offerings and the acquisition of new clients. Many of the participating SMEs could not identify the exact product extensions, stating only that they had strong intentions in this area and that product extension initiatives would be client-led. In relation to new client acquisition objectives, the majority of the participating SMEs had clearly identifiable targets. Gaining repeat business from existing clients and business process management are the final two areas that are generally reported as having high priority objectives. The majority of participating SMEs report strong targets in relation to gaining repeat business from existing clients, while business process management objectives tend to be more diverse in nature – some SMEs intend to improve the sales process while other SMEs have an objective to change the deployment licensing model for their software products.

[10] The hierarchy of business objectives pyramid was constructed by analysing the responses from all participants to each question in the business objectives survey instrument. Both the frequency and the strength of reported objectives are jointly considered, with more frequent and/or stronger objectives being placed higher on the pyramid. The tiering of the pyramid was a natural outcome of the clustering that was evident in the data, i.e. the analysis demonstrated that there were points in the data where relatively large gaps were observable in the frequency and/or strength of the reported business objectives.

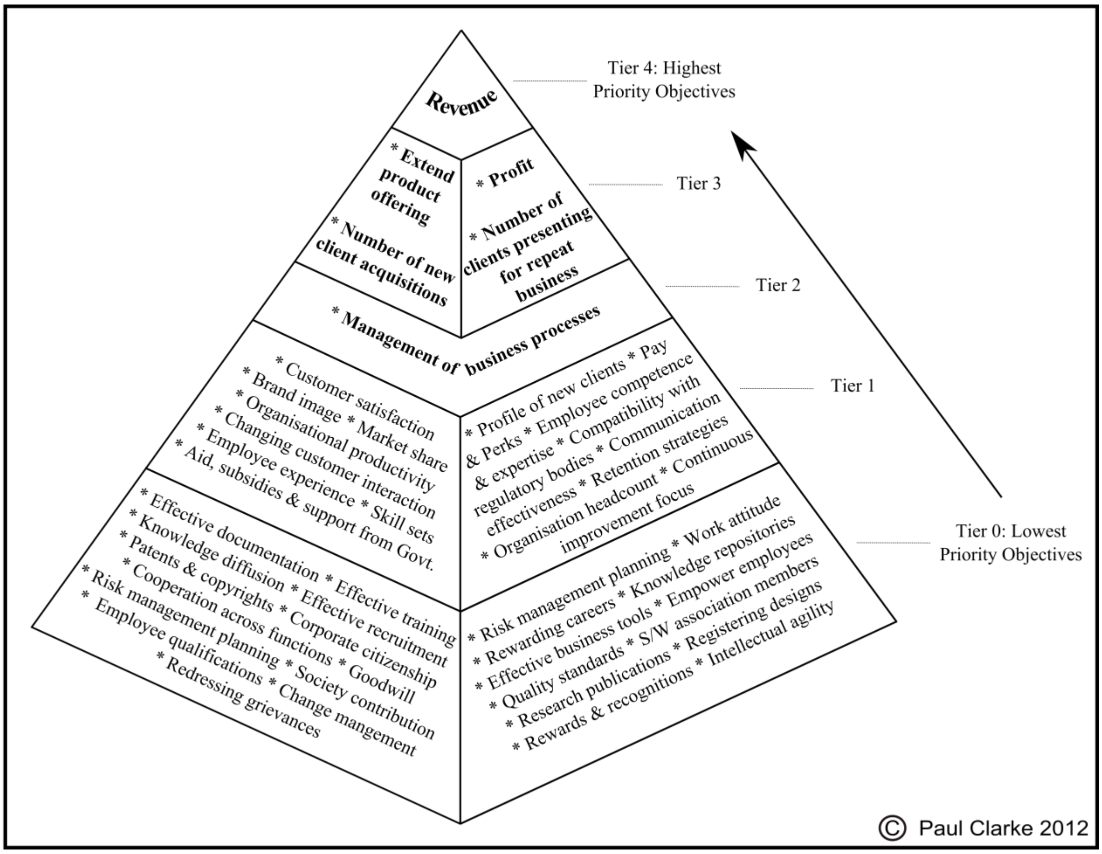


Figure 28 Hierarchy of business objectives for software SMEs

8.7.3.2 Low priority business objectives

The evaluation of the business objectives data also reveals that there are a number of areas where software SMEs have low priority objectives. Most notable among these objectives are: *contributing to society*, and *redressing grievances* which essentially did not feature for the any of the participating organisations. The absence of objectives in these areas may be related to the survivability concerns which can exert a significant pressure on the business as a whole. There was also a strong indication from the participating SMEs that they do not intend to invest in training programmes and that they essentially have no objectives with respect to seeking or retaining a recognised quality standard. SMEs can therefore be characterised as organisations where best practice models are only implemented where their absence is considered to be a barrier to sales development, thus confirming the findings of earlier studies (Coleman and O'Connor 2008). Furthermore, in software SMEs, training is *"on the job"* and there is very little interest in pursuing research publications.

8.7.3.3 General observations concerning business objectives

In addition to identifying the high and low priority objectives for software SMEs, there were also a number of additional interesting observations. Very few of the participating companies manage risks in an organised or systematic way and they have no plans to start formalising risk management. Risk management is one dimension of self-reflection and is a conduit for continuous improvement – therefore, SMEs might derive some of the benefits of continuous improvement by establishing a risk management discipline. It was also interesting to discover that several of the participating SMEs held the view that maintaining existing levels of customer satisfaction was going to be difficult if the business was to expand – since the small number of existing clients were presently receiving very high levels of dedicated support.

The participating SMEs also report that other than *"on the job"* skills development, there is very little focus on career development for staff and that career growth was not considered to represent a high priority objective for the business. Furthermore, there appears to be little interest in (or possibility of) retaining underperforming employees. These findings are somewhat at odds with the theoretical high importance of knowledge workers in software development – where continued career development may lead to increased motivation and higher retention rates among staff members. A further interesting observation was made in relation to the patenting ambitions of the participating SMEs, where only a few of the organisations have expressed patenting targets. The general belief among the participating SMEs was that patents are very expensive to file and that they offer little protection for the technology. For those SMEs that are engaged in patenting, the principal reported benefit is the protection of the valuation of the company for investors or purchasers.

8.7.3.4 Business objectives with greatest degree of achievement

Following the completion of the second phase of the business objectives examination, it was possible to analyse the data to examine which objectives are consistently being reported as being fully achieved. Equally, it was possible to identify those objectives which the participating companies were less successful in achieving. Figure 29 outlines the results of the analysis of the achievement of business objectives.[11]

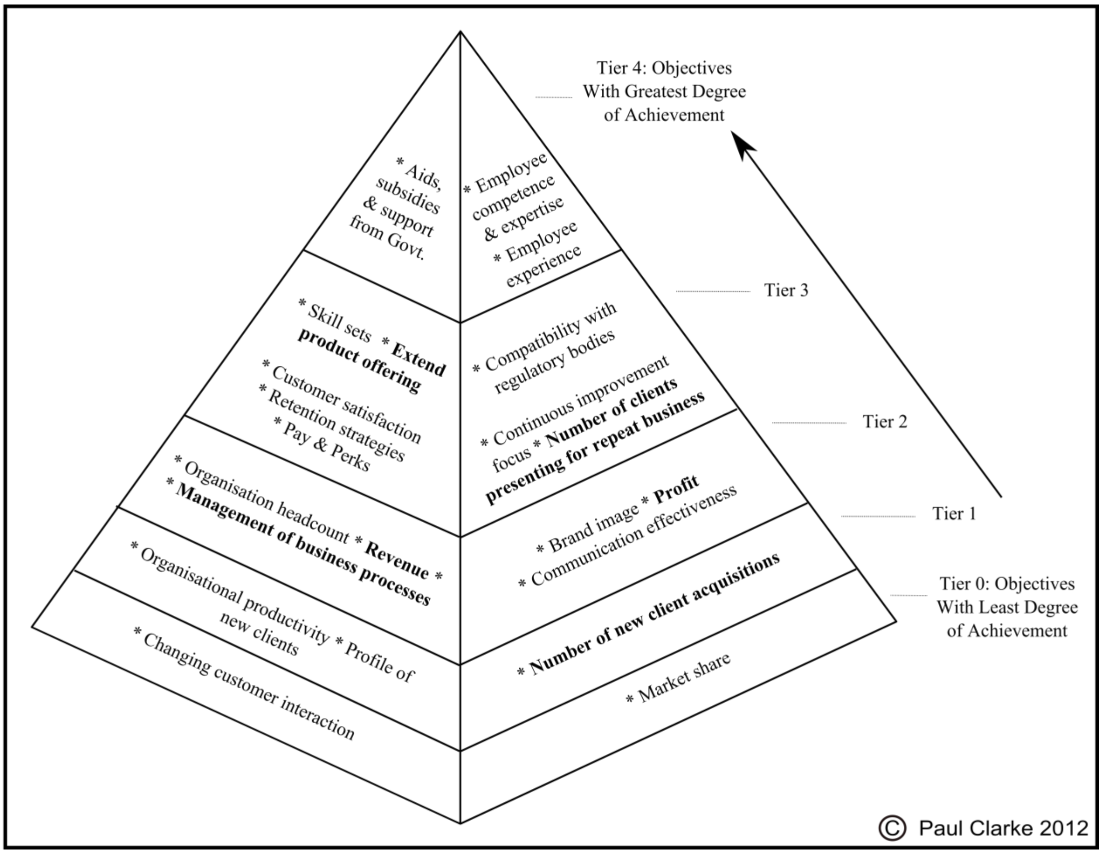


Figure 29 Hierarchy of business objectives achievement in software SMEs

An analysis of the business objectives achievement data demonstrates that there are 11 objectives that software SMEs tend to be most successful in achieving. The highest degree of achievement was in respect of the objectives in relation to obtaining aids, subsidies and support from government. This finding was slightly surprising, as in the first phase of the investigation, some of the participating organisations had highlighted that they often had difficulty in obtaining financial support and assistance from government. The analysis also revealed that the software SMEs were quite

[11] Objectives in Tiers 1-4 of the hierarchy of business objectives for software SMEs (Figure 28) are considered in this analysis. The objectives that are highlighted in bold in Figure 29 are those that were in the top three tiers of the hierarchy of business objectives for software SMEs (Figure 28). The hierarchy of business objectives achievement pyramid was constructed by analysing the responses from all participants in relation to the achievement of the earlier recorded objectives. The average reported degree of achievement for each given objective was calculated across all organisations, with objectives with higher average achievement values being placed higher on the pyramid. The tiering of the pyramid was a natural outcome of the clustering that was evident in the data, i.e. the analysis demonstrated that there were points in the data where relatively large gaps were observable in the average achievement values for the business objectives.

strong at achieving their objectives in relation to employees, specifically in relation to the competence, expertise and experience of staff. Additionally, the data analysis demonstrated that the software SMEs are relatively successful when it comes to achieving business objectives in the area of compliance with regulatory bodies. In the case of the participating organisations, a number of the individual companies operate in business domains wherein regulatory compliance is a pre-requisite for business – for example, certain telecommunications and confidential data processing systems. Therefore, it is not altogether surprising to discover that the participating SMEs report that in general they have been successful in terms of satisfying the regulatory bodies associated with their business domain (since a failure to satisfy regulatory considerations could have an immediate negative effect on the organisation).

The participating software SMEs also report considerable success when it comes to extending their product offering. Many of the participating organisations had explicit new features and capabilities that were identified as objectives from the first phase – and in most cases, the participating organisations were successful in implementing the features or products. Of all the high priority objectives identified in the first phase (those objectives on Tiers 3 and 4 of the hierarchy in Figure 28), the participating companies were most successful in terms of implementing new product base or in enhancing existing product base.

The analysis also demonstrates that among the participating organisations, companies were reasonably successful in terms of their objectives in relation to customer satisfaction levels and gaining repeat business from existing clients. These two objectives would appear to be related, since if a client is satisfied, they are also more likely to present for repeat business. Furthermore, the participating companies report that they tend to be relatively successful in terms of achieving objectives related to employee retention strategies, pay and perks.

8.7.3.5 Business objectives with least degree of achievement

While the participating companies were most successful in areas such as obtaining aids, subsidies and support from government, and in terms of extension to product offerings and generating repeat business from existing clients, there are a number of other areas in which the objectives were not as successfully met. Notably, the companies were not quite as successful when it came to meeting revenue and profit targets, or in terms of objectives in relation to the business process. The broad view that can be established from the analysis is that software SMEs work hard at retaining and extending business with existing clients, but that other aspects of their business objectives are more difficult to realise. In addition to the difficulty associated with achieving revenue and profit objectives, the participating SMEs were less successful again when it comes to hitting targets for new client acquisitions. These observations highlight some interesting aspects of the software SME sector.

Firstly, none of the participating organisations were listed on a stock exchange and therefore, they are not subject to the predictability of revenue and profit targets that are generally demanded by the markets. As a result, aggressive revenue and profit targets may be set by small company owners – since there is no immediate negative funding impact from failure to achieve objectives. Or perhaps it is the case that there is not a great deal of oversight of the original financial objectives with a view to tempering them against the unrealistic targets set out by the principle agent, the owner. It should also be highlighted that small software development companies are often involved in market creation and innovative product development – the results from which can be difficult to predict in advance.

Secondly, small businesses are like any other general type of business in one key respect – in that it tends to be less difficult and less costly to obtain new business from existing clients than it is to secure entirely new clients (Osterwalder and Pigneur 2002). Once a relationship is established and trust is in place, it is more likely that a customer will be prepared to do business with an established supplier who has existing delivery experience. This particular issue could be exacerbated in software SMEs that are trying to convince potential new clients of the benefits of their innovative new product – they may first have to work hard to create the market for the new product, something that is acknowledged as being a significant challenge (Molina-Castillo and Munuera-Aleman 2009).

The data analysis also reveals that there was a low degree of success in terms of achieving objectives concerning the profile of new clients. This may further emphasise the dependence that software SMEs can have on existing clients, and the difficulty that they can experience when trying to attract new customers, especially if the profile of the desired customer base is divergent from the existing customer base. Therefore, small companies may be much more exposed to the demands of a few key clients – something that is further evidenced by the relatively low degree of success that the participating organisations had in terms of changing the nature of customer interaction. One final observation in relation to the difficulty of securing new clients is the fact that the participating organisations were overwhelmingly falling short of the market share objectives that were captured during the first phase engagement.

8.8 Extended data analysis

In this section, the statistical relationship between SPI activity and business success is examined, initially visualising the relationship using scatter graphs and later calculating correlation coefficients. The latter part of this section analyses the role of situational change in terms of how it relates to SPI activity and business success.

8.8.1 SPI activity and business success

The relationship between any two variables can be illustrated using a scatter graph. A scatter graph is a graph with a scale for each variable and upon which variable values are plotted in pairs, with the basic intention being to visualise whether there is any pattern among the points (Harper 1991). One of the key concerns of this study is the examination of the relationship between business success and SPI activity. Therefore, scatter graphs are initially used to visualise the collected data, so as to get a sense of the relationship (or non-relationship) between SPI activity and business success. In Figure 30, the two interpretations of business success, *basic* and *WNM*, are plotted along with SPI activity in scatter graphs.

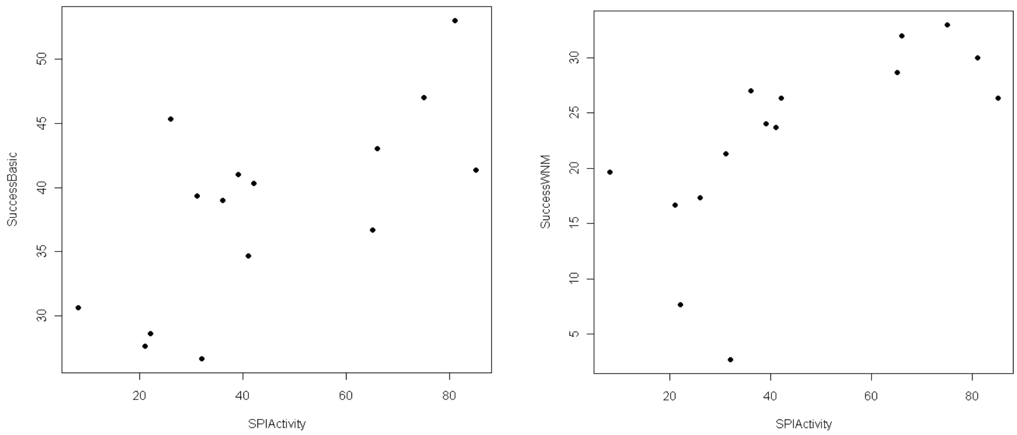


Figure 30 Scatter graphs for SPI activity with basic and WNM business success

In the case of both scatter graphs in Figure 30, increases in SPI activity appear upon visual inspection to be presenting alongside increases in business success. For example, it can be seen that in general increased SPI activity is tending to present alongside increased basic success. Similarly, higher levels of SPI activity are tending to present in cases where there is also higher WNM business success. Observations such as these indicate that there is a positive linear relationship between business success (both basic and WNM) and SPI activity. However, although scatter graphs permit a visualisation of patterns in the data, they are limited insofar as they lack a measure of the closeness of the relationship between the plotted variables (Harper 1991). Correlation analysis provides a measure of the closeness of the relationship between variables (Reilly 1997) and therefore, the next step in the analysis is to perform some standard statistical correlations on the data.

In order to express quantitatively the extent to which two variables are related, it is necessary to calculate a correlation coefficient (Haber and Runyon 1980). Since this study is non-interventionist in nature, the utilisation of correlation is appropriate for the research. Had the research instigated or recommended particular SPI actions for the participating organisations (with a view to measuring the effect of the actions), then it is probable that a different type of experimentation would be required (for example, utilising control groups). As discussed in more detail in section 3.6, one of the major weaknesses of correlation is that it does not permit the examination of causation. As a result, correlation coefficients alone can only be used in an inductive or persuasive fashion – and ideally, in conjunction with a suite of other considerations. A number of different correlation coefficients exist, and when choosing an appropriate correlation technique, a number of factors should be considered, including (1) the type of scale of measurement in which the variables are measured, and (2) the nature of the distribution of the underlying data (Haber and Runyon 1980).

This study predominately uses ordinal measurement scales, such as the Likert scales in Table 8, Table 10, and Table 12. On an ordinal data scale, the numbers that are assigned to variables express a *greater than* relationship, but they do not indicate *how much greater* (Kranzler 2003). Therefore, points on an ordinal scale indicate a ranking but they do not necessarily have equal units of measurement. This is different from interval or measurement scales (Koopmans 1981) (for example, using a metre scale to measure height) where the numbers that are assigned to variables reflect relative merit and have equal units of measurement. In the absence of equal units of measurement, it is advisable to use a ranked correlation coefficient, such as the Spearman coefficient (generally designated as *R*) (Hinton 1995). Therefore, the Spearman coefficient is employed as the primary correlation coefficient for this study. To further increase the validity of the correlation analyses, an alternative ranked order correlation technique, the Kendall rank coefficient, is also employed as an additional measure. In general, Kendall coefficients tend to produce broadly similar, if slightly lower, absolute coefficients to the Spearman coefficient and there is no strong reason for preferring one over the other (Colwell and Gillett 1982).

While it is generally recommended that ranked correlations are applied to ordinal data (Hinton 1995), such as the data utilised in this study, there are claims that it is permissible and useful to perform interval data correlation techniques on ordinal data (O'Brien 1979), especially in cases where the underlying data is normally distributed (the distribution for the variable is bell-curved in shape (Harper 1991)). Using Q-Q plots in R, the distributions for SPI activity, basic and WNM business success were all found to correspond to generally normal distributions. Therefore, it is permissible to perform interval data correlations on the three variables. The most common interval data correlation technique is the Pearson product-moment correlation co-efficient (generally designated as *r*) (Kranzler 2003), and in view of the increased precision associated with product-moment correlation coefficients (Harper 1991), Pearson product-moment correlations are also conducted on the study data. Using both ranked order and product moment correlation techniques in tandem will allow for a more thorough assessment of the data.

The possible values of correlation coefficients range from +1.00 to -1.00. The closer the correlation coefficient is to +1.00 or -1.00, the closer the relationship between the variables; and the closer the correlation coefficient is to 0, the less close the relationship (Harper 1991). One further important aspect of correlation coefficients concerns the probability value (*p-value*). The p-value indicates the odds of a chance occurrence (Hinton 1995), or in other words, the probability of an error when making declarations in relation to the significance of the correlation coefficient.

The Spearman, Kendall and Pearson correlation coefficients and associated p-values were calculated for the SPI activity and business success data sets collected in this study. The results from the correlations, which were cross-checked in *STATISTICA* and *R*, are presented in Table 16.

		SPI Activity & Basic Business Success	**SPI Activity & WNM Business Success**
Spearman	R	0.66	0.81
	p-value	0.009	0.0002[12]
Kendall	T	0.49	0.61
	p-value	0.011	0.0015[12]
Pearson	r	0.68	0.68
	p-value	0.005	0.005

Table 16 Spearman, Kendall and Pearson correlation coefficients

Examining the correlation coefficients for business success (both basic and WNM) and SPI activity, it is found that the coefficients range from 0.49 to 0.81 (refer to Table 16). This indicates that there is a positive correlation between SPI activity and business success. Additionally, when taking the sample size, the correlation coefficients and the p-values into consideration, it can be declared that the correlation coefficients are statistically significant (refer to Hinton (1995) for a breakdown on the

[12] Spearman & Kendall correlation p-values cannot necessarily be reliably computed where a variable has the same value for two separate cases. In the case of this study, the participating organisations *Cameron* and *Michelin* both got a WNM Business Success score of 26.33. Hence, the Spearman and Kendall p-value calculations are marginally compromised when correlating SPI Activity and WNM Business Success.

interpretation of significance). Therefore, the analysis of the data collected in this study supports the first research hypothesis:

> ***H1:*** *SPI activity is positively associated with business success in software SMEs.*

Since the correlations between SPI activity and business success range from 0.66 to 0.81, the evidence gathered in this study suggests that these two phenomena are not just positively correlated, but they are in fact tending towards being strongly positively correlated. At 0.81, the Spearman correlation coefficient for WNM business success and SPI is strong. This correlation is particularly interesting, as the WNM interpretation is considered by the researcher to present the fairest interpretation of the relative business success in the participating organisations (as discussed in Section 8.2).

Additional variations on the correlations presented in Table 16 were also conducted on the data in order to perform a sensitivity analysis on the results. For example, it is possible that there are areas of overlap across some aspects of the HSC-based business process objectives and ISO/IEC 12207 processes; including such areas as risk and knowledge management. Although the scope for overlap is relatively small, all potential instances of overlap were removed from the business objectives data and the correlations were re-performed on the resulting data suites. The results of these additional correlations demonstrate that SPI activity and business success continue to be positively correlated even when the potentially overlapping areas are removed from the HSC data (these additional correlations produce statistically significant Spearman and Pearson correlation coefficients ranging from 0.56 to 0.75). The following section analyses the data to investigate if the degree of situational change has an influence on the relationship between SPI activity and business success.

8.8.2 The influence of situational change

Although the earlier analyses have demonstrated a positive association between business success and SPI activity, it is important to examine the influence of situational change. In a situation where there is little recorded change, the need for SPI is potentially reduced. Therefore, an organisation with a low level of situational change should be capable of achieving relatively high levels of success with modest levels of SPI activity. It should also be expected that in general, organisations that present with relatively large amounts of SPI as compared with their particular degree situational change, are tending to present with greater business success. In Figure 31, the relationship between SPI activity, situational change, and basic business success is presented, with Figure 32 plotting the relationship between SPI activity, situational change, and WNM business success.

Prior to elaborating on the discussion in relation to the influence of situational change, it is important to highlight that the measurements that are being utilised are non-parametric and therefore, not of equal units of measurement. Furthermore, they are measuring quite different phenomena (SPI activity is a different phenomenon to situational change). However, the amount of SPI activity implemented in one organisation can be compared with the amount of SPI activity in another organisation, since it is the same phenomenon that is under investigation and it is measured using the same technique; the same case applies for the other phenomena under investigation. However, when analysing the amount of SPI activity in an organisation as compared with the degree of situational change in the same organisation, in effect two different types of measurement are being compared. This raises the concern that such comparisons are not useful – that apples and oranges are being compared. However, since the quantification of the individual phenomenon is considered to provide a relative measure of the phenomenon in the different organisations – SPI activity in one organisation can be compared with SPI activity in a second organisation, just as situational change in one organisation can be compared with situational change in a second organisation – this section proceeds on the basis that these relative measures are useful for an analysis of the different phenomena within a single organisation. However, it is important to stress that the individual measures are not measured on the same scale – even if the values in the graphs are presented on the same scale.

In Figure 31, a number of clear observations can be made regarding the role of situational change. Firstly, from visually examining Figure 31, it can be seen that where SPI activity exceeds the degree of situational change, organisations are tending to be more successful. Three of the top four basic business success organisations match this profile: *When*, *Colleran* and *LordHenry* This suggests that in order to be more successful, it is important for SPI activity to be aware of the degree of situational change and to respond in a proactive fashion, or in other words, for the amount of SPI activity to be relatively high when compared with the degree of situational change. Further evidence for such a relationship can be observed in the case of the organisation *Cameron*. Although *Cameron* records the single highest amount of SPI activity in the study, it is not the most successful company. However, *Cameron* also records the highest degree of situational change, such that the amount of SPI activity is not particularly large relative to the degree of situational change.

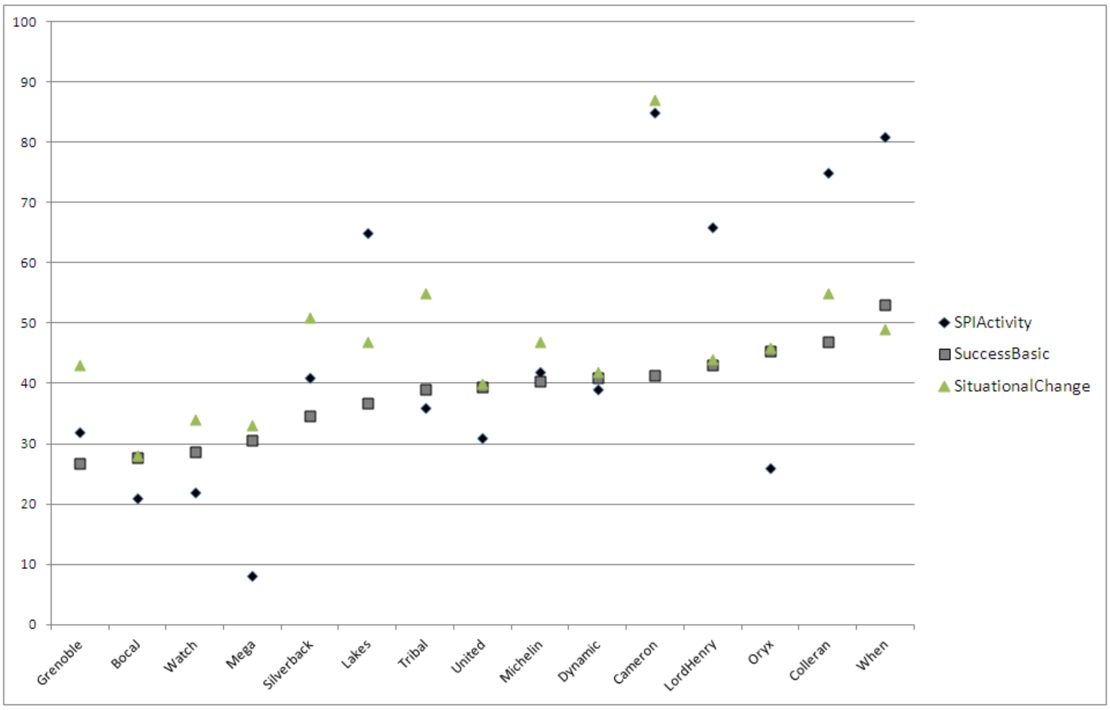


Figure 31 SPI activity, situational change and basic business success

The second major observation that can be made in relation to situational change is that three of four worst performing companies (under the basic business success interpretation) register the three lowest degrees of situational change: *BocaJ*, *Mega* and *Watch*. Although the awareness of situational change was not measured in this research, this second observation may suggest that organisational awareness of situational change or the creation of changing situations has an influence on business success. For example, an organisation with a lower awareness of situational change may be less inclined towards SPI, with the result that business success prospects are limited. In this respect, it is interesting to note that the three companies reporting the lowest levels of situational change are also reporting the three lowest amounts of SPI activity. However, since awareness of situational change is not measured in this research, these particular suggestions cannot be substantiated in this study – and other factors could account for the observations. Returning to the second major observation, under the alternate view of business success, WNM business success, three out of the five worst performing companies (*BocaJ, Mega* and *Watch*) register the lowest degrees of situational change (refer to Figure 32). Whatever the explanation for this observation, it would appear on the basis of the data collected in this study that organisations that report relatively lower levels of situational change are also tending to report relatively lower levels of business success.

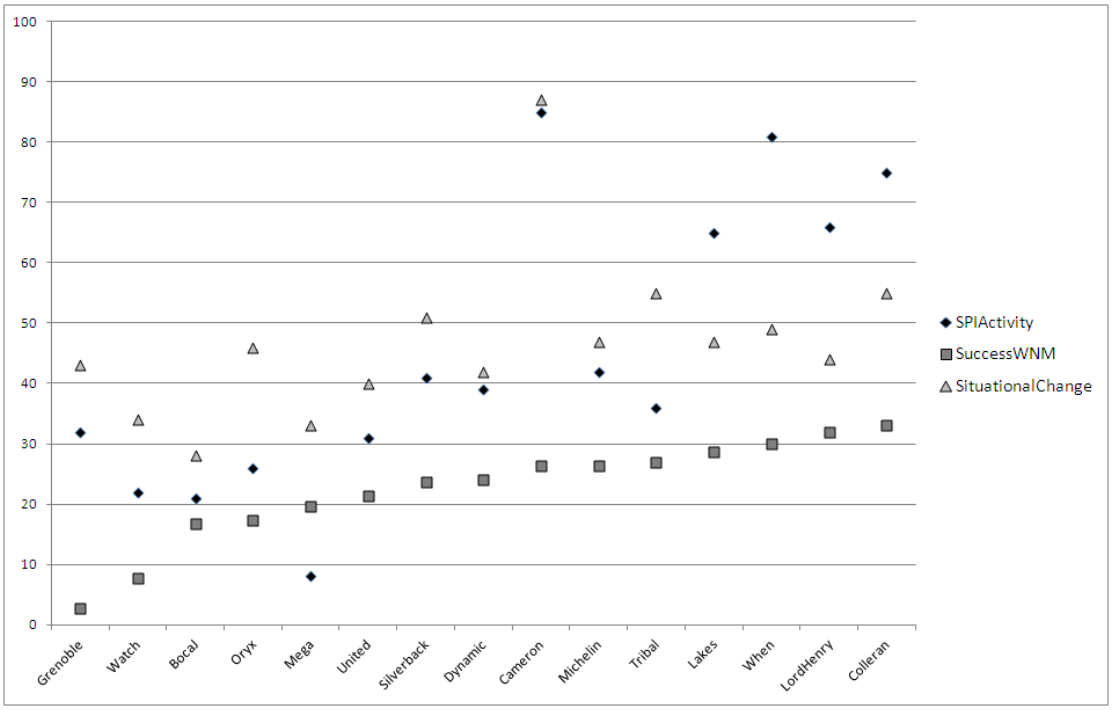


Figure 32 SPI activity, situational change and WNM business success

The WNM success perspective (refer to Figure 32) further illustrates the first observation from the basic business success perspective: where the amount of SPI activity is relatively high when compared with the degree of situational change, the prospects for business success appear heightened (a point that is discussed in more detail in the following chapter). In Figure 32, it can be seen that each of the top four most successful organisations have a profile wherein the amount of SPI activity exceeds the degree of situational change – and these are the only four organisations in the study to present with such a profile. Six additional organisations with comparable levels of situational change but with lower levels of SPI activity (*Tribal, Michelin, Dynamic, Silverback, Oryx* and *Grenoble*) fail to achieve as much business success as the top four organisations.

There is a further indication in the data that the degree of situational change is an important consideration when implementing SPI. The two organisations with the largest gap between SPI activity and situational change are *Mega* and *Oryx*. These two organisations achieve relatively low levels of business success under the WNM interpretation – with both organisations falling into the lowest five performing companies. This observation further supports the case for situational change being an important consideration for SPI activity. However, when taken in conjunction with the other observations identified in this section, it is not permissible to reach a definitive conclusion in relation to the role of situational change. On balance, however, the evidence does suggest that the second hypothesis may be true (though much additional data and corresponding analysis would be required in order to make an absolute judgement in relation to H2):

> ***H2:*** *To maximise business success, SPI activity should be in proportion to the degree of situational change.*

8.9 Data validity

This section discusses the issues related to data validity and the steps taken in this research to increase the level of confidence in both the underlying data and the conclusions. While data validation is not explicitly included as a distinct phase in the research process adopted in this study (refer to Section 1.5), it is an integral component of the research model. The purpose, therefore, of this section is to consolidate and highlight the various data validation steps that were implemented throughout the study (many of which have already been indirectly identified in the relevant sections of this thesis). Furthermore, this section is included so as to explicitly identify any threats to validity introduced by data validation concerns.

8.9.1 Reference frameworks

Initially, this research identified three distinct phenomena that required observation: (i) SPI activity, (ii) situational change, and (iii) business success. In order to observe any phenomenon, it is important that a reliable reference framework is identified. Such a framework should explicitly outline all of the major dimensions of the phenomenon, such that the framework can be harnessed in an effort to quantify or observe the phenomenon in practice. If the framework chosen to investigate a phenomenon is deficient (i.e. some major aspects of the phenomenon were absent in the framework) then the quality and validity of the collected data would be undermined. Therefore, this research endeavoured to utilise the most extensive and/or most widely accepted reference frameworks for each of the phenomena under examination. While there are drawbacks to aiming at such extensive reference frameworks, such as the increased time and complexity required for study design, and data collection, analysis and evaluation, the utilisation of comprehensive reference frameworks is the first step in raising the validity of the data utilised in this study.

In the case of SPI activity and business success, the selection of ISO/IEC 12207 and the HSC (respectively) represented comprehensive frameworks for each of these phenomena. In the case of SPI activity, a more comprehensive or more generally accepted software development process reference framework does not exist. In the case of the HSC, it is the only published business scorecard based approach that is explicitly designed for use in software development companies. Furthermore, the HSC is an extension of the most widely utilised business success scorecard approach, the BSC. Therefore, the HSC does represent a credible and comprehensive framework from which to investigate business success.

The HSC, however, is a relatively recent publication and there is no existing evidence to suggest that is it widely used at present in order to examine business success in software companies. Therefore, the HSC could potentially be deficient in terms of coverage and general acceptance and as such, this introduces a threat to the validity of the data collected in respect of business success. In order to mitigate this threat, the HSC based survey instrument that was developed and employed in this research includes two questions to elicit any objectives that may have been overlooked by the

HSC based questions. Specifically, the first question inquires about any additional financial-related targets that were not identified. The second question, which was included at the end of the survey instrument, explicitly inquires about any additional business objectives that were not addressed in the course of the survey instrument.

In the case of situational change, no published comprehensive reference framework of the situational factors affecting the software development process was found to be in existence. Therefore, it was incumbent on this research to develop a broad framework of the situational factors affecting the software development process. The developed framework was systematically constructed such that the impact of individual underlying data sources is clearly recorded and presented in this thesis (refer to chapter 5). Therefore, the situational factors reference framework is verifiable and extendible, both of which characteristics increase the general quality of the framework. Furthermore, the situational factors reference framework is considerably broader in scope (with 170 individual sub-factors) than any individual related framework. Hence it is the most appropriate reference framework that exists for investigating the extent of change in the situational factors affecting the software development process.

An additional benefit of the situational factors framework is that it has been subject to peer review in a leading journal: Information and Software Technology (Clarke and O'Connor 2012). During the peer review process, a number of related works were suggested for inclusion in the framework development. However, upon detailed analysis, the proposed works were found to be adequately addressed in the situational factors reference framework as it was initially presented. The results of this analysis exercise are reproduced in Appendix E. As a result, additional confidence regarding the scope and content of the situational factors reference framework can be drawn from the experience of the peer review process for the related publication. However, despite the benefits noted above, the situational factors reference framework is not without its limitations. For example, the framework lacks consensual validation and as such, may be limited in terms of scope or design. Furthermore, this is the first and only study to utilise the framework. As a result, the data collected in relation to situational change may be compromised insofar as it may overlook important considerations. These limitations introduce a threat to the validity of the data collected regarding the degree of situational change recorded in this study.

8.9.2 Development of survey instruments

Having determined the most comprehensive and effective reference frameworks for investigating the phenomena of interest, the next major challenge relates to the development of corresponding survey instruments. From a validity perspective, it was important that the survey instruments would accurately and honestly reflect the underlying reference frameworks. In this respect, the survey instruments employed as much of the detail from the underlying standard as was practically dischargeable. Furthermore, certain steps were introduced in order to maximise the confidence in the validity of the survey instruments.

The first step involved the systematic development of the survey instruments. In the case of the SPI activity survey instrument, the entire ISO/IEC 12207 standard was studied in detail – with all of the processes, activities and tasks being explicitly extracted from the standard. Much of the accompanying text in the standard was extracted and later, in the second step, the SPI activity survey instrument was slowly developed from the extracted data (refer to section 4.4). At this time, detailed records of the SPI activity survey instrument development were retained for future reference or audit. A third step to maximise the validity of the survey instrument involved subjecting draft versions of the survey instrument to independent experts review (with subsequent feedback integrated into later revisions of the survey instrument). Furthermore, a fourth step involved piloting the survey instrument with an industry partner in order to further raise the confidence that the survey instrument was fit for purpose. A final step involved the addition of an explicit question to challenge the scope of the survey instrument, wherein participants were requested to identify any additional software development processes that were not present in the survey instrument.

While all of the steps identified above were performed in order to maximise the validity of the data that would be extracted during the data collection, and although the survey instrument clearly identifies all of the core processes in ISO/IEC 12207, the SPI activity survey instrument does not contain the extensive and in-depth level of detail that is native to ISO/IEC 12207 – this would be prohibitively expensive and impractical to discharge (since ISO/IEC 12207 is 138 pages in length). Therefore, a further threat to validity concerns the gap between the details contained in ISO/IEC 12207 and the depth of coverage offered in the SPI activity survey instrument.

To maximise the validity of the data collected for business success and situational change, similar steps to those outlined in the previous paragraphs were introduced in the creation of the corresponding survey instruments (refer to section 5.7 and 6.3). Both of these survey instruments reflect the topology of the underlying reference frameworks, both were subject to review, both contain questions that inquire as to additional objectives or situational factors beyond the scope of the survey instrument, and both were piloted with an industry partner. However, as with the SPI activity survey instrument, owing to practical considerations there exists a gap between the extensive detail of the underlying reference frameworks and the depth of coverage offered in the survey instruments. Although the collected data was rich in detail, the existence of this gap introduces an additional threat to data validity – though this threat is not considered to be significant in nature and the survey instruments are considered by this author to be fit for purpose.

8.9.3 Discharge of survey instruments

Having identified appropriate reference frameworks and corresponding survey instruments for the phenomena under investigation, the next task was to discharge the survey instruments. Since the survey instruments were discharged only by one researcher (the author of this thesis), it was not deemed necessary to independently validate the questions in the survey instruments to ensure that each question was commonly understood by multiple interviewers. While the language used in the survey instruments was of a generally well understood nature (especially with respect to the components of the software development process and also with regard to business success objectives), there is a possibility that some subjectivity was introduced on the part of the participants concerning the exact understanding of various different aspects of the phenomena under investigation. Furthermore, the subjectivity or bias of the interviewer could potentially influence the subjects – to minimise this threat, formal validation of the survey instruments would have been beneficial. However, this threat is not considered to be large as the interviewer discharged the survey instruments without going into the detail of the broader research project. For example, the interviewees were not aware of the research hypotheses or questions.

The subjectivity of individual participants is also a concern regarding the details provided in relation to the extent of process change, business success and situational change. Essentially, what one participant might consider as *minor* in nature, a second participant may consider as *moderate*. The Likert scales employed in this research are four-point – *none*, *minor*, *moderate*, and *significant* for SPI activity and situational change; *not achieved*, *partially achieved*, *mostly achieved*, and *totally achieved* for business success. These delineations are considered to be sufficiently well understood by participants in order to get a representative view of the phenomena in the different organisations – however, they are not perfectly and identically understood by all participants (nor can they be) and therefore, along with the differences in meaning surrounding the underlying language and concepts, a threat to data validity in introduced by these considerations. This threat could have been partially addressed through broadening the number of participants from each organisation, perhaps through the adoption of focus groups. Alternatively, this threat could have been minimised by requesting that interviewees explicitly described the reported process change, situational change or business objective – though such a process would have required a significantly larger time commitment from study participants. However, this research did not implement such steps to address potential participant subjectivity or bias, and consequently participant bias or subjectivity present as a threat to data validity.

In addition to subjectivity concerns, a related – if more sinister – consideration relates to the honesty of the participants. In this respect, there was never any indication that the participants were being dishonest or deliberately misrepresenting the situation. The motivation for the participants for dishonest reporting was considerably reduced by the introduction of participant and organisational anonymity in the study. Indeed, in some cases, multiple individuals from a single organisation participated independently in this research; hence further reducing the possibility for a concerted misrepresentation. Furthermore, this research is observational by design, with no participant involvement at a stakeholder level in the research. Perhaps if the research had been interventionist in nature, there may have been a temptation on the part of participants to overstate or understate the true extent of SPI, situational change or business success. This particular temptation may also have increased owing to the selection of participating organisations from a network of known contacts (which was unavoidable from a practical perspective). Therefore, even though there is no suspicion of false or biased reporting on the part of participants, it remains a possibility that some of the interviewees may have deliberately misrepresented some details, and consequently, false or biased reporting presents as an additional threat to the validity of the data.

8.9.4 Data entry

Having collected the data from the participants, it was important that the data was carefully and reliably entered into the applications that would be used to conduct the data analysis. In the first instance, *Microsoft Excel* was adopted as the initial data analysis application. In order to minimise threats to data validity, all of the data was checked for correctness on two separate occasions – having entered the data into *Microsoft Excel*, each item of underlying data was checked and subsequently, re-checked. In addition, a random sample of the survey scripts were later selected and again re-checked to ensure that the data was correctly entered. Therefore, it is considered highly unlikely that any errors were introduced during the data entry phase and as such, there is at most a negligible threat to the data validity resulting from data entry.

8.9.5 Data analysis

The data analysis involved the use of multiple applications. Firstly, the data entered into *Microsoft Excel* was subject to various summations and formulae. In order to minimise the threat of errors in the summing or manipulation of the data, all of the core measurements were subjected to automated checking. This automated checking involved the parallel and independent calculation of the various measures, with the different results being automatically cross-checked in a separate table in *Microsoft*

Excel. Where two values that were expected to be identical were discovered to be in contradiction, the cross-check table would highlight the values in a bold red font. This cross-checking introduced a significantly increased level of confidence in the validity of the data.

In addition to *Microsoft Excel*, *R* and *STATISTICA* were employed in order to perform the data correlations in relation to SPI activity and business success (refer to section 8.5). Just one of these additional applications could have been utilised to calculate the required statistical data. However, so as to maximise the validity of the data, two separate applications – with two completely independent data entry mechanisms – were employed. By adopting this approach, a very high degree of confidence can be developed in relation to the advanced statistical analyses. Although the utilisation of two separate data analysis applications created an additional workload for the researcher, the prize of increased confidence in the statistical findings is considered to be worthwhile in view of the additional time invested. As a result of utilising two distinct and independent data analysis applications, the correlation analysis is not considered to introduce a threat to data validity.

To further ensure that the data correlations were not erroneous, misrepresentative or deficient in scope, three separate data correlation techniques were employed (including both parametric and non-parametric analysis): Pearson product moment, Spearman and Kendall rank order (refer to section 8.8). Furthermore, all three correlation techniques are integrated into the data analysis and discussion regarding the statistical relationship between SPI activity and business success. As a result, the findings in relation to the statistical correlations are not considered to introduce a threat to validity. Of course, the field of statistics is very large and multiple statistical tools and techniques can be employed in the analysis of data. However, one of the primary purposes of this study was to examine the relationship between SPI activity and business success, and statistical data correlation offers a very strong fit in terms of co-relating or associating these two phenomena.

In terms of analysing the role of situational change, this thesis has presented graphical representations of the three phenomena under examination in the study (refer to section 8.8.2). The representations plot SPI activity and situational change alongside basic and WNM business success. The discussion that is presented along with the graphical representations demonstrates that there is evidence in the data that suggests that H2 may also be valid. The future elicitation of larger pools of data and the subsequent application of more sophisticated data analysis techniques would help to more substantially analyse the role of situational change. However, and as is elaborated upon in section 10.8, future research investments into situational change should first concentrate on validating the reference framework of the situational factors affecting the software development process (as opposed to increasing the pool of organisations that have been subjected to the corresponding survey instrument).

8.10 Summary

This chapter started by identifying the approach to quantifying business success, SPI activity and situational change, after which an overview of the data quantification was presented. As well as highlighting the most and least common areas for SPI activity and situational change, this chapter also profiled the business objectives for software SMEs. The latter part of this chapter examined the relationship between SPI activity and business success, finding that increased SPI activity is positively associated with business success in the participating software SMEs. Additionally, the role of situational awareness was discussed, with the data collected in this study suggesting that in order to maximise the business success prospects, the amount of SPI activity should be relatively high when compared with the degree of situational change. The final part of this chapter presented a detailed discussion in relation to data validation and threats to validity. The following chapter examines the meaning, impact and importance of the data analyses outlined in this chapter.

9 Data Evaluation

9.1 Introduction

This chapter evaluates the findings of the data analysis. Initially, the importance of the observed positive association between SPI activity and business success is explored. Thereafter, the influence of situational change is discussed, followed by an examination of the nature of SPI and situational change in software SMEs – with a focus on parallels with earlier research and on new insights afforded by this study. Finally, both ISO/IEC 12207 and the HSC are examined in terms of their relevance to software SMEs.

9.2 Relationship between SPI and business success

As outlined in chapter 8, this study employed standard correlation techniques to examine the strength of the relationship between SPI activity and business success. The study findings indicate that increases in SPI activity are positively associated with business success. While members of the software process and SPI communities may not be surprised by these findings, the literature review presented earlier in this thesis suggests that in practice, companies may not place a high priority on the software process, with software SMEs only implementing process improvements in response to negative business events. Since the evidence provided in this study indicates, for the first time, that there is a positive correlation between increases in SPI and increases in business success, software SMEs may begin to place a higher priority on SPI.

The hypotheses set out in this study essentially seek to examine the relationship between increased SPI activity and business success in software SMEs. Since general business process management theory and practice inform us of the importance of maintaining effective business processes (McCormack and Johnson 2001, Skrinjar, Bosilj-Vuksic and Indihar-Stemberger 2008, Kaplan and Norton 1992, Neely, Adams and Kennerley 2002), and seeing as the needs of many businesses are continually changing, it should therefore be expected that software SMEs with a stronger SPI focus should, in general, tend to be more successful. Nonetheless, it is worth briefly examining the possibility that it was the increased business success that gave rise to increases in SPI activity.

In this study, those companies that reported relatively higher levels of business success were also reporting relatively higher amounts of SPI activity. However, if we consider that SPI is concerned with improving the software development process to best support the business needs, then SPI should not just be implemented in companies presenting with higher levels of business success - we should expect to also observe evidence of increased levels of SPI activity in some of the companies that are performing poorly; since a decline in the fortunes of an organisation is just as likely to be a catalyst for process change. Such evidence is weak in this study, with the general observable trend being that companies with relatively lower levels of success also having relatively lower levels of SPI activity (refer to Figure 30). However, the evidence is not conclusive in this respect, with for example, two of the lesser successful organisations (under the WNM interpretation: *Grenoble* and *United*) reporting moderate levels of SPI activity when compared with the other participating organisations.

Another interesting finding from the study is that although different amounts of SPI are being performed in the participating organisations, without exception all of the participating software SMEs implemented some SPI activity over the study period. While some of the organisations only performed small amounts of SPI, the fact that all of the participating organisations performed some degree of SPI suggests that the software process is not fixed but rather that the process in continually changing, even if only modestly. This particular finding supports the argument that no one size fits all when it comes to software development processes (Boehm and Turner 2003) and that process adaptation is an important activity for software development companies (Poulin 2007). If we accept that the software development process is continually changing in organisations, then valuable future research should focus on profiling the impact of situational change on aspects of the software development process. In particular, it would be useful to better understand the interplay between situational changes and software process changes, perhaps developing a relational mapping between aspects of the software development process and aspects of the (changing) situation. Such a relational mapping would identify which aspects of the situational context affect which software development processes – and to what extent.

9.3 The important role of situational change

In chapter 8, the influence of situational change was analysed, finding that the degree of change in the situational factors affecting the software development process would appear to be an important consideration when implementing SPI in software SMEs. The idea that organisations need to continually adapt their processes in response to changing situational contexts has received explicit attention in the field of economics. In economics, the *neoclassical* theory of the firm presents profit maximisation as the reason for existence (Cyert and Hedrick 1972). However, other theories of the firm (that can be used in conjunction with the neoclassical theory) have emerged over time and one of these, the *evolutionary* theory of the firm (Nelson and Winter 1982), appears to describe many of the characteristics of software development organisations.

According to the evolutionary theory of the firm, a company is constituted by both physical and human assets, and there is a strong focus on organizational capability (Chandler 1992). Organizational capability refers to the knowledge, skills and experience within the company and can essentially be thought of as the "*integration of individuals' specialized knowledge*" (Grant 1996 p375). With knowledge, skills and experience accumulating over time, it is the ability to continually and effectively learn that gives rise to the dynamism that will ultimately propel the organisation to success (Chandler 1992). The firm, therefore, is promoted as "*a locus where competencies are continually built, managed, combined, transformed, tested and selected*", where the vital consideration relates to how "*new knowledge [is] materialised in new competencies*", and where "*a lock-in to inefficient routines*" (Cohendet et al. 1999 p227-8) is perceived as a major threat to a company's prospects. Consequently, a dynamic capability to transform routines is considered to provide a basis for competitive advantage (Cohendet et al. 1999).

The evolutionary theory of the firm also describes the need for spare capacity to enable the application of the accumulated knowledge, the absence of which would result in a lack of evolution. In essence therefore, the evolutionary theory of the firm suggests that organisations that are better at learning, that can utilise their spare capacity to promote adaptation and innovation, are more likely to be successful and to endure. Such innovation can be broad in scope (including such items as technical and process innovation) and although closely related to business success, innovation itself is not within the direct scope of this research. The evolutionary theory of the firm indicates that companies seek to gain a strategic advantage through "*continuous product, process and organisational innovation*" (Jacobson D. and Andreosso-O'Callaghan 1996 p43) and that this is achieved by ensuring that flexibility exists at all levels in the company. The fundamental perspective is that "*improvement is always possible and ideas for improvement can come from everyone*", with the premise that firms "*depend upon learning to maintain a competitive advantage*" (Best 1990 p13).

Since they continually create new software and often depend on innovation as a cornerstone of their business, software development companies would appear to offer a particularly good fit for the evolutionary theory of the firm. Software companies exist in a fast moving, innovative environment and don't have self-contained, slowly evolving products that are pushed off the manufacturing line but rather, the product is continually evolving. Therefore, there is a compelling need for software development organisations to continually apply accumulated experience and knowledge to work practices. Although he does not use the term dynamic capability, Watts Humphrey recognises the importance of learning and process improvement stating that "*reactive changes generally make things worse… [and] crisis prevention is more important that crisis recovery*" (Humphrey 1989 p21). Evolutionary theory offers similar wisdom, promoting the importance of metamorphosis via learning within an organisation, for both the products and the production process.

The software development process constitutes a significant and complex component of a software development business, and therefore, when viewed through the evolutionary theory of the firm lens, the success of a software business will be affected by the degree to which the organisation is capable of adapting the software development process. It would therefore appear sensible for software development companies to measure their SPI activity while at the same time being cognisant of the degree of situational change – so as to have insight into the extent to which they are learning and evolving. Earlier research has reported that an organisation's ability to optimise the development process provides a better approach than traditional audits

(Poulin 2007) and that when determining the most appropriate software development process, "*the bottom line…is that you must know yourself (as a company, team or individual) and the diversity of existing processes out there, and adopt and adapt what's best for you*" (Lindvall and Rus 2000 p18). Therefore, this researcher considers that situational context and changes to the situational context are critical considerations for software development organisations – and the initial reference framework of the situational factors affecting the software development process developed by this research (refer to chapter 5) is an important new resource to assist the examination of individual settings.

As outlined in chapter 8, the data collected in this study suggests that the evolutionary theory of the firm applies to software SMEs: the most successful organisations were those where the amount of SPI activity was relatively high when compared to the extent of situational change. In effect, the framework and conceptualisation of the evolutionary theory of the firm may help to explain the data gathered in this study.

9.4 Nature of SPI in software SMEs

An in-depth examination of the data reveals that this study has collected valuable new information in relation to the nature of SPI in software SMEs. These new insights raise interesting challenges for established process maturity reference frameworks and quality management standards. Prior to elaborating on these insights, the parallels with earlier SPI research are first presented.

9.4.1 Parallels with earlier research

Some of the findings from the data analysis confirm the findings from earlier related studies. For instance, the study finds that many of the participating organisations are making improvements to the general area of requirements management, including capture, sign-off and change management. This is not surprising, as a number of earlier studies have highlighted the importance of improvements in the areas of requirements management for software SMEs. For example, in the SPI in Regions of Europe (SPIRE) project (Sanders 1998), one of the participating organisations, *Cunav Technologies*, derives benefits from prioritising the improvement of their requirements analysis process. In another related study, it is recommended that software companies should focus on improving requirements capture as an important area for SPI (Keane and Richardson 2005). Keane and Richardson (2005) also suggest that it can be important to focus SPI on the estimation process – another observation that is echoed in the findings of this research. Improvements to aspects of testing are commonly being reported in this study, confirming reports from earlier studies that improvements in the testing process have positive benefits for software SMEs (Nikitina and Kajko-Mattsson 2010).

In this research, the very nature of the reported SPI activity is also revealing. As well as structured and formal SPI, SPI activity is also detected at an informal level, with significant evidence that aspects of the process are evolving in a manner that may be invisible to traditional process assessments. This observation relates to tacit or informal SPI, and it confirms the findings of earlier studies (Basri and O'Connor 2011, Fuller 2001, Ryan and O'Connor 2009). In this study, such tacit knowledge led SPI is particularly evident in the language that participants used in responding to some of the questions. These responses included statements such as: we're now *"more aware of"*; aspects of the process are *"discussed further"* (as opposed to being documented or formalised); there *"might be a little bit more discussion [in relation to]"*; we're *"certainly thinking about [it]... more than last year"*; that *"we're more conscious of [it]"*; that some parts of the process are *"more of a concern"*. This type of response from participants indicates that organisations are improving their awareness of the problems that they face and that this is leading to improvements in the way that they address aspects of their work. However, this type of improvement in work practices is not evident in additional artifacts or records, and therefore, probably invisible to contemporary SPI frameworks.

Tacit knowledge led SPI also takes the form of extensions to mental checklists. For example, one organisation reported that *"we're probably more concerned about the maintenance so, when we're in code, we will try and carve out all of the redundant stuff just because we want to reduce the maintenance burden"*. There is no formal change that could be recorded, just that they are now somewhat more mentally aware of the need for refactoring. Since this mental awareness is giving rise to improved refactoring in practice, this represents an improvement in the process of software construction. However, since this is not formally implemented in practice, and there are no records or measurements in relation to the improvement, it is likely to be overlooked by conventional process improvement assessments. While not central to the hypotheses under consideration in the study, this particular observation represents an important new contribution in its own right.

9.4.2 New Insights

No earlier published software SME based study attempted to quantify the amount of SPI being implemented in practice, both formal and informal, across the full spectrum of software development processes. Therefore, this research represents a new approach to examining SPI, an approach which has proven to be valuable in terms of identifying that there is a considerable degree of SPI being implemented in software SMEs, with some of the SPI being informal and minor in nature.

The study presented herein takes the broad range of software specific and system context processes into consideration, and investigates both the small informal process changes and the larger, more formal instances of SPI. Taking this approach to examining SPI, this study establishes evidence of wide variety and intensity of SPI across the broad range of software processes (refer to Table 13). The findings show that software SMEs initiate a considerable amount of SPI, albeit in varying quantities in different organisations. This finding is both intriguing and valuable as it suggests the software development process is being continually evolved in software SMEs. Furthermore, the data from this study demonstrates that the software development process in software SMEs is largely evolving through a series of minor or moderate adaptations. This can be considered to be a type of tweaking of the process rather than fundamental reshaping of the process. Or as one organisation described some of their changes: *"there has been some very minor changes… but nothing fundamental"*.

The discovery in this study that there are many minor and informal process improvements taking place in software SMEs has implications for established process maturity frameworks. In such frameworks, the top level of maturity is concerned with optimising, an activity that is intended to ensure that the software development process is continually harmonised with emerging needs, knowledge and technologies. However, such frameworks are largely not implemented in software SMEs, and even if they were to be implemented in SMEs, experience from larger organisations suggests that very few companies would progress to highest level *optimising* stage. In this research, the data collected and analysed suggests that software SMEs are implementing a wide variety of SPI even though they do not implement established process maturity frameworks. This recorded SPI could be the result of process optimisation as a principle and therefore, software SMEs may have embraced the optimising principle of the highest maturity level of established process maturity reference frameworks, while at the same time largely ignoring the staged process maturity concept in general.

Although much of the process change is minor or moderate in nature, the extent of the recorded change was surprising. At the outset of the study, the researcher did not expect to record such extensive SPI in practice in software SMEs – and no earlier study had suggested that this might be the case. The findings reveal that the software development process landscape is constantly evolving in software SMEs. One of the participants asserts that "*it doesn't really matter what you have formally because tomorrow it is all going to change"*, with another company stating that *"everything that we've changed… will be changed again"*. In addition to this high degree of process change, this study also finds that software SMEs espouse an *a la carte* approach to process methodology adoption. One organisation states that *"we don't formally follow the methodologies… [that] as a business, we're not capable of fully following one process"*. As a result the company reports that, in terms of implementing software development processes, *"we've kind of cherry picked a bit"*.

While there are dangers to cherry picking in this manner, such as implementing an agile process without any refactoring, this finding is also of interest as it indicates that when it comes to the software development process, there is no one prescribable solution. Each organisation must implement and adapt their software development process in tandem with their needs, environment and resources. Or as one software process researcher states, *"it is reasonable to assume that the optimal process is not static but is organization-dependent and time-dependent, and will have to be modified as the context in which the organization operates evolves"* (Poulin 2007 p25). The evidence collected in this study supports this assumption.

While the ability to mix and merge different development methodologies offers some benefits for software SMEs, the study also highlighted the danger of unbridled process methodology manipulation. For example, in relation to code refactoring, one of the participating organisations reported that *"we try to do more refactoring but getting management agreement for refactoring that doesn't produce any new features is a difficult challenge"*. This particular comment highlights the struggle with limited resources in smaller organisations. Software SMEs can be fighting for existence (Von Wangenheim, Anacleto and Salviano 2006) and as a result may try to maximise short term value at a cost to long term maintainability. In this sense, there can be a temptation to promote the attractive value-led features of agile software development approaches (Fowler and Highsmith 2001) while demoting the important supporting agile practices. Of course, in the long term, such decisions could have highly undesirable outcomes, with the possibility of ending up with a code base that is expensive to support and maintain.

The nature of the process change reported in the study is also worth further discussion. In addition to the tacit knowledge process changes identified earlier, the type of process changes and the views of the study participants indicate that there is some considerable degree of ambiguity in practice as to what constitutes SPI. In one instance, an organisation reports that an aspect of the process is unchanged, but that *"the change I would say is that we have gotten better"*. In this case, the respondent was referring to the initiation of whiteboard sessions to discuss the implications of new requirements. In another case, an organisation reports that *"we do a project post-mortem meeting, we always did, but not to the same level that we do now"*. While the post-mortem meetings and the whiteboard sessions may have happened to some minor extent in earlier periods, they are considered to have increased in frequency (and probably also in detail). Since the process is generally neither documented nor policed in software SMEs, an increased frequency of post-mortem meetings or whiteboard sessions is difficult to classify. In a more formally defined environment, this could be classified as increased process adherence, however, in the absence of a clear definition of the development process, this increased tendency towards white board sessions and post-mortem meetings is considered to be best classified as a process improvement in its own right – though very much informal in implementation.

Many of the participating organisations also reported that they had adopted new supporting tools. Six of the organisations adopted completely new tooling for aspects of their software development. In some cases, these tools related to improved capability to visualise and track requirements and change requests, hence improving communication and awareness. In other cases, the new tools were adopted in order to improve the process for product deployment out to clients. In another company again, a hosted cloud-based application was used to improve the ease of document sharing with clients. Such evidence suggests that software SMEs continually integrate new tooling solutions to aid the task of software development. In one concrete example of tool adoption improving the task of software development, one of the participating organisations started to use the same tool for both source code control and live deployment of web based applications. Historically, no single tool could accomplish both of these tasks, and therefore, this tooling advancement allowed the organisation to improve the efficiency of its software development and deployment activities. With six of the participating organisations reporting that they adopted new tooling to support aspects of their software development, this study finds that the adoption of improved tooling can be important for software SMEs – and this particular finding is not evident in earlier published works.

While new tooling solutions can offer improved efficiencies, the study demonstrated that there can be confusion regarding the role of tools in SPI. One organisation reports that *“the electronic tracking system that we’re using is being used a lot more often now”*. The suggestion is that the increased usage of the tool is not representative of a process improvement. However, increased utility of a tracking system is representative of at least improved process adherence, and could possibly be considered to be an instance of process improvement. While changes to the utilisation of supporting tools (or to the tools themselves) can sometimes represent instances of SPI, the findings of this study suggests that software SMEs don’t consistently distinguish between instances of SPI and instances of tooling improvements.

The findings of this research also indicate that organisations desire to become more independent from the demands of individual customers and that they seek to become more strategic in their actions. For instance, one company reported that the selection of requirements has become "*less market driven"* resulting in *"more breathing space"*. A second organisation comments that they are *"trying to move away from the question 'can we' and more towards 'should we'"*, and in so doing they consider that they are *"becoming more capability led and less market led"*. A third company commented that *"over the past year… the business got a clearer focus on what it was trying to build product-wise whereas before, there could be a new idea every week"*. While this study was not focused on examining why such changes were occurring, the nature of the feedback suggests that the companies consider that a degree of liberation from the whims and demands of customers is generally viewed in a positive light. This may be especially true when an increased focus can be devoted to independent, strategic product decision making. In their early years, some software SMEs might struggle for survival and may be over-extended in terms of developing a market or retaining one or more critical business accounts. Perhaps after this period of passed, software SMEs become more stable and viable businesses, with the result that they are more able to manage the sometimes unrealistic demands of customers.

9.5 Nature of situational change in software SMEs

There are a number of features of the data analysis in relation to situational change that highlight the particular challenges faced in software SMEs. Perhaps the most notable among these challenges is the rate of growth or decline in headcount over the year under investigation. Each of the organisations witnessed some change in headcount, which is to some extent to be expected. However, it is the rate of change in headcount that is striking. Firstly, two of the participating organisations witnessed a reduction in headcount of 40% or greater - losing 15 out of 35 employees in one case, and 8 out of 20 employees in the second case. Furthermore, 9 out of the 15 participating organisations experienced headcount growth of 25% or higher, with 6 of these 9 organisations growing their headcount by 50% or more. In some cases, these percentage increases are in part accounted for by the fact that the organisations were very small at the start of the study. However, in other cases, quite a large number of new personnel were introduced to the participating company. Since changes in headcount have a significant effect on the process of work, on the basis of the data collected in this study, it would seem that software SMEs are continually challenged with process issues resulting from changes in the headcount of the organisation.

Another significant challenge faced by software SMEs relates to the rate of increase in the volume of transactions that are processed by their software products. Ten of the participating organisations report increases in the volume of transactions. Some of the organisations note that their "*traffic continues to grow"*, while for others the increase is *"very significant"*. Further evidence of this growth includes reports that *"storage requirements are growing"* and that an increasing database size *"is one of the biggest challenges that we have now"*. Increased throughput and storage requirements can place a large burden on already stressed applications, and increasing throughput and storage capacity in a system is often an area that will require special and intensive attention. However, software SMEs don't necessarily have a great deal of bandwidth in terms of addressing non-critical activities. Therefore, the challenge of implementing either temporary or permanent process improvements to address ever-increasing volume related issues is likely to be very difficult for many software SMEs.

The issue of ever-increasing volume throughput and storage issues is further exacerbated by the ever-increasing demand for improved performance. For example, 12 of the participating organisations report an increased required performance in their product(s) over the period under investigation. One of the companies reports that the performance requirement has *"gone up by 20%",* while another responds that performance has *"increased probably by 30%"*. Another company again reports that their product *"has to run twice as fast".* Some of the other organisations report that the increased performance requirement is *"significant"*, that it is "*always increasing",* and that *"the customer is always demanding more fast and more reliable [products]".* This increased demand for higher performance is a significant challenge for the software SMEs in the study and it represents a significant stress for the participating companies – especially when coupled with the increased data requirements discussed in the previous paragraph.

In addition to the challenges already noted for software SMEs, this research finds that the supporting technologies are themselves continually changing. Eight of the participating organisations reported that they adopted new technologies over the year under investigation. Some of the organisations report changes to the programming related environment, including languages, compilers and associated tooling. Other companies report that they had to support additional operating systems, including traditional desk top operating systems and emerging mobile device operating systems. The adoption of new technology requires effort, with one respondent stating: "*we've started using several different technologies over the past year and people have had to skill up on them and share their knowledge among the team"*. And the adoption of new technologies is not a whimsical step on the part of the organisation but it is an effort to identify a competitive advantage – as another respondent states: *"there's new technologies all the time… we're constantly trying to find something better".*

In this study, the participating organisations reported a whole host of changes to the situational contexts. The most frequently reported changes relate to (often significant) changes in headcount, increasing data management and speed of execution requirements, and the shifting sands of technology. Taken together, these situational changes make for a hugely challenging software process management proposition.

9.6 Relevance of ISO/IEC 12207 to software SMEs

As outlined in chapter 4, the SPI survey instrument utilised in this study was based on ISO/IEC 12207. As a result, the survey instrument adopts the same language and process classification that is presented in ISO/IEC 12207. A closing question was incorporated into the SPI survey instrument so as to provide the participants with an opportunity to comment on the completeness of the SPI activity survey. This question encouraged participants to identify any process improvements that they had adopted that were not addressed in the survey instrument.

Overwhelmingly, the participating organisations asserted that they considered the survey to cover all of the areas related to their software development process – which is further evidence that ISO/IEC 12207 is comprehensive in scope and thus appropriate as the software process reference framework for this study. The participants stated that *"it's been pretty comprehensive"*, *"you've pretty much covered the entire software lifecycle"*, and *"I think you've gone through the full lifecycle"*. Since ISO/IEC 12207 is developed and maintained by an international panel of software development experts, it is not surprising that the study participants find that the survey instrument is comprehensive in terms of addressing the software development process. However, the participants do identify some drawbacks associated with the application of ISO/IEC 12207 in the setting of software SMEs.

There are some indications that the language adopted in ISO/IEC 12207 is not easily accessible in software SME settings. One of the participants reported that he found the language to be *"almost awkward"*, with another participant stating that the questions relate to *"very formal mechanisms"*. It is true that ISO/IEC 12207 is a large, valuable and formally developed resource but perhaps the extent of the international standard and the associated formality render the standard unsuited to software SMEs. Just as earlier studies have indicated that software SMEs consider process maturity frameworks to be infeasible rather than non-beneficial (Staples et al. 2007), perhaps ISO/IEC 12207 is similarly infeasible for software SMEs. Whether or not ISO/IEC 12207 is infeasible for software SMEs, in practice ISO/IEC 12207 has very little exposure in this sector – with none of the participants having previously been aware of the existence of the standard.

With ISO/IEC 12207 being such a comprehensive and carefully constructed software lifecycle process reference framework, it seems regretful that it is not utilised in smaller software development organisations. Although ISO/IEC 12207 is designed to identify the atomic processes required for software development, the evidence of this study indicates that there are some significant language and stylistic differences when compared with the software development process as practiced in software SMEs. Clearly, with over 400 process tasks, ISO/IEC 12207 is likely to be considerably beyond the scope of individual software SMEs. However, some gaps exist in the language and concepts. For instance, some of the terminology which forms part of the vernacular of software SMEs (and many larger software development organisations) doesn't feature anywhere in ISO/IEC 12207; for example, *refactoring* and *timeboxing*. This particular observation resonates with a broader issue – that there is a lack of a generally accepted dictionary of terms for software development. Software development is a complex affair, with many participants and numerous tasks. Therefore, the absence of a generally available and generally accepted dictionary of terms can give rise to considerable ambiguity and inconsistent use of terminology.

9.7 Relevance of HSC to software SMEs

For the purpose of this research, the HSC has provided a comprehensive framework from which to assess the objectives of software SMEs. In chapter 8, a hierarchy of the business objectives for software SMEs was presented, with six key areas identified as being particularly important for business objectives: *revenue*, *profit*, *extension of product offerings*, *new client acquisitions*, *repeat business from existing clients*, and *business process management*. In addition to the HSC-based questions, participants were expressly asked if there were any objectives that were not covered as part of the interview, and consistently they reported that the interview was comprehensive – with comments such as *"quite comprehensive"*, *"it's a fairly comprehensive framework"* and *"good questions"*. However, there are some indications that the scope of the HSC may in fact be overly-broad for the purpose of examining software SMEs. Furthermore, a number of additional objectives were identified. Therefore, equipped with the data analysis and evaluation from this research, there are a number of recommendations that can be made with respect to the use of the HSC as a reference framework for future research in the area of business success for software SMEs.

The initial business objectives interview required on average ninety minutes with a senior manager from each of the participating SMEs, and later interview transcribing required a minimum of six hours per interview. This is a time consuming process for both the interviewee and the interviewer. Furthermore, the bulk of the HSC business objectives featured as relatively low priority items for the software SMEs in this study. Indeed, one of the interviewees commented that there was *"a lot of emphasis on objectives which certainly in a small company doesn't ring true… [that] we've got revenue and product type objectives, other than that we tend to sort of blow with the wind a little and react, rather than being overly pro-active in the sense of setting any particular targets."* Therefore, the first recommendation is as follows:

> **Recommendation 1**. If a future study of business objectives in SMEs were to use the HSC (or the HSC-based survey instrument produced by this research), the researchers could consider removing or consolidating the objectives that are in the lowest tier of the hierarchy in Figure 28.

While recommendation 1 could be adopted by a future research effort in the software SMEs sector so as to make the identification of objectives more efficient, it is considered important to retain closing questions that permit the interviewee to comment on any additional objectives. It is difficult for any survey instrument to be absolutely complete and the inclusion of such closing questions permits the elicitation of objectives that are beyond the scope of the survey instrument or that have possibly been overlooked. In the application of the HSC-based survey instrument to software SMEs, such closing questions allowed the researcher to discover a number of additional objectives that are not native to the HSC. Consequently, the second recommendation is that questions in relation to a number of additional objectives should be included in the survey instrument:

> **Recommendation 2**. Future research into the business objectives in software companies of all sizes should include questions relating to objectives in the areas of (1) financial liquidity (sometimes termed cash flow); (2) off-shoring or outsourcing some aspects of the development work; (3) mergers and acquisitions (M&A).

9.8 Summary

This chapter evaluated the study data, highlighting the positive association between SPI activity and business success. Thereafter, the important role of situational change was explored, followed by an evaluation of the nature of SPI and situational change as identified in this study. Finally, the relevance of ISO/IEC 12207 and the HSC to software SMEs was examined. The next section reviews the overall study, revisiting the research hypotheses and questions, and examining the research contribution and the impact on the field.

Part 5 Conclusions

The final part of the thesis contains chapter 10 which summarises the research. Initially, the research hypotheses and questions are revisited, after which the research contribution and impact are examined. Finally some research limitations are noted along with suggested areas for future research.

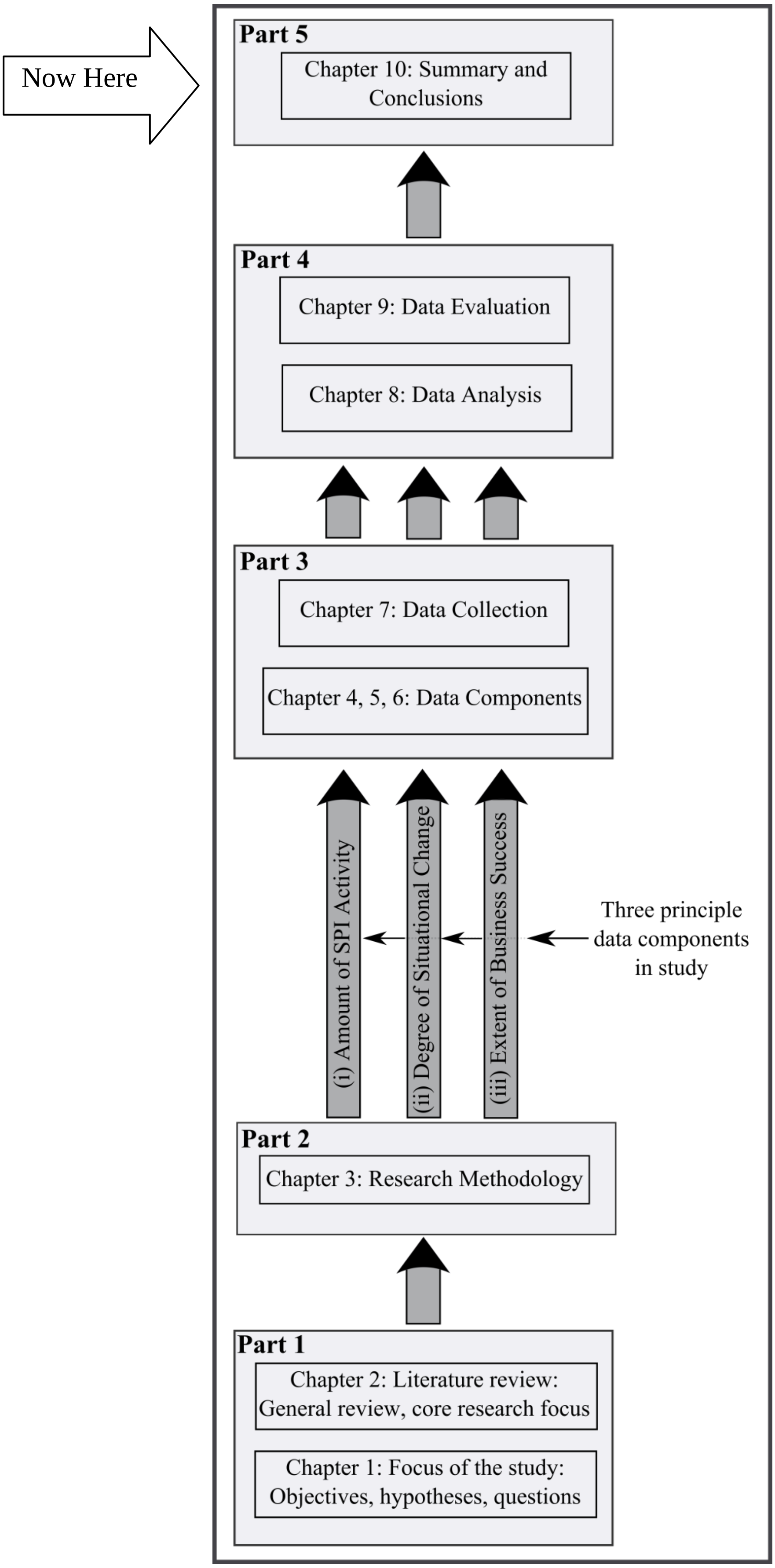


Figure 33 Map of Thesis – Part 5

10 Summary and Conclusions

10.1 Introduction

This chapter commences by providing a brief overview of the central considerations and motivations in this study. Thereafter, the research hypotheses and questions are revisited in light of the data collected, analysed and evaluated in this study. The primary and secondary contributions are also presented, along with a discussion of the impact that the contributions have on the field of software development. This chapter concludes by highlighting some of the limitations of this research, along with recommendations for future research.

10.2 Overview

Business processes are generally accepted as having a significant role to play in supporting business success. Although some debate exists regarding the degree of importance of business processes in enabling business success (Vergidis, Tiwari and Majeed 2008), various empirical studies have demonstrated that business processes and business process improvement are important for business performance (McCormack and Johnson 2001, Skrinjar, Bosilj-Vuksic and Indihar-Stemberger 2008). Furthermore, the importance of business processes is acknowledged by the inclusion of business process performance perspectives in many of the contemporary business performance measurement frameworks (Kaplan and Norton 1992, Neely, Adams and Kennerley 2002).

In software development organisations, the software development process is a large and complex component of the broader business process landscape. Indeed, for many software development companies, the software development process is likely to be both the largest and the most complex process. Therefore, the effective management and maintenance of the software development process is important for business success. However, the findings from relatively recent studies suggest that software SMEs can have a low software development process priority (Baddoo and Hall 2003), tending to only implement SPI in response to negative business events (Coleman and O'Connor 2008). Clearly, therefore, there is a degree of incongruence concerning the theory and practice of SPI in software SMEs.

At the conceptual stage, this research set out with the objective of explicitly and systematically examining the relationship between SPI and business success in software SMEs. This is the first study to undertake such an examination and given the noted incongruence between theory and practice with respect to the implementation of SPI in software SMEs, this alone was considered to offer sufficiently important scope for research. However, at an early stage in the research conceptualisation, it became apparent that the implementation of SPI should be related to some extent to the degree of situational change that is occurring in a software development setting. Earlier research efforts highlighted the importance of such contextual considerations and as a result, a decision was taken to explore three distinct phenomena in a number of participating software SMEs: (1) the amount of SPI activity; (2) the degree of situational change; and (3) the extent of business success.

A review of the related literature confirmed that ISO/IEC 12207 and the HSC provided comprehensive reference frameworks for the examination of the amount of SPI activity and the extent of business success. However, and although many research works refer to the importance of situational context with respect to the software development process, no comprehensive reference framework of the situational factors affecting the software development process had previously been published. As a result, it was incumbent on this research to develop a new reference framework of the situational factors affecting the software development process.

Having identified appropriate and comprehensive reference frameworks, the next step involved the creation of a series of survey instruments that could leverage the contents of the reference frameworks to the task of examining the three primary phenomena of interest. The survey instruments were carefully and systematically developed from the reference frameworks, with subsequent expert reviewing and piloting employed in order maximise the effectiveness and utility of the instruments. Thereafter, considerable effort and industry succeeded in securing the participation of fifteen software SMEs, a task which was aided by the extensive industrial experience of the researcher. Over a sixteen month period, the survey instruments were deployed to the participating organisations, the data from which was subjected to extensive data analysis and evaluation.

Equipped with the outputs from the data analysis and evaluation, this concluding chapter revisits and examines the original research hypotheses and questions, with a key focus on outlining the major contributions of this research. Some limitations are also outlined and finally, some suggested areas for future research are discussed.

10.3 Revisiting the research objective and hypotheses

As outlined in chapter 1 of this thesis, the objective of this study was to investigate if SPI is having a positive influence on business success in software SMEs. On the basis of the evidence gathered and analysed in this research, it would appear that, in practice, SPI is positively associated with business success in software SMEs. In order to examine the relationship between business success and SPI in detail, two hypotheses were created for exploration in this research:

> ***H1:*** *SPI activity is positively associated with business success in software SMEs.*
>
> ***H2:*** *To maximise business success, SPI activity should be in proportion to the degree of situational change.*

Using the empirical evidence gathered in this study, hypothesis 1 has been proven: increased SPI activity is positively associated with business success in software SMEs. This is the first study that was explicitly designed from the outset to examine the relationship between SPI and business success in software SMEs, and it is the first study to provide empirical evidence of the positive relationship between the broad spectrum of SPI and full range of business success considerations in software SMEs.

In the case of the second hypothesis, the data collected in this research is insufficient to definitively prove or disprove the proposition. However, and as discussed in detail in section 8.8.2, the evidence gathered in this study does suggest that in order to maximise business success, SPI activity may need to be in proportion to the degree of situational change. In particular, organisations that present with a relatively high level of SPI activity when compared with the recorded degree of situational change are also presenting as being the most successful organisations in the study. This is first time that such evidence has been accumulated and it suggests that situational context may be an important consideration when implementing software processes and SPI.

10.4 Revisiting the research questions

In addition to the two research hypotheses discussed above, three research questions were also developed in order to further understand the three primary phenomena under investigation:

> ***RQ1:*** *What is business success for software SMEs?*
>
> ***RQ2:*** *What aspects of the software development process are commonly undergoing SPI in software SMEs?*
>
> ***RQ3:*** *What aspects of the situational context are commonly experiencing change in software SMEs?*

In the case of RQ1, it was established that software SMEs have a number of high priority business objectives in the areas of: *revenue*, *profit*, *extension of product offerings*, *new client acquisitions*, *repeat business from existing clients*, and *business process management*. Furthermore, and as outlined in detail in chapter 8, there are a number of areas wherein software SMEs tend not to have business objectives, especially with respect to *contributing to society*, and *redressing employee grievances*. A hierarchy of business objectives for software SMEs was developed from the data collected in the participating companies and this is summarised in Figure 28. Therefore, RQ1 is considered to have been answered in considerable detail.

In the case of RQ2, it was established that the most commonly reported process areas for SPI are: the requirements analysis process, the process for provision of an infrastructure for development, the process for tendering, bidding and negotiating with clients, the software installation process, the project planning process, the software construction process, and the configuration management process. Furthermore, a comprehensive hierarchy of SPI in software SMEs was developed, as presented in Figure 26. Therefore, RQ2 is considered to have been comprehensively answered.

In the case of RQ3, it was established that the most common areas for change in the situational context are: changes to organisational headcount, changes in technology, increases to the required ease of product installation and operation, increases to the numbers of transactions to be processed by products, and general increases in required product performance. Furthermore, a broad set of additional aspects of situational change were reviewed at length, leading to the development of a hierarchy of the situational changes in software SMEs – as presented in Figure 27. Consequently, RQ3 has also been answered in considerable detail.

10.5 Research contribution

This research has made contributions in a number of important respects. Taken collectively, the contributions represent a significant addition to the SPI, software SME and business success bodies of knowledge. In this section, the primary and secondary research contributions are discussed.

10.5.1 Primary contributions

Firstly, this is the first published research to demonstrate using empirical data that increases in SPI activity are positively associated with business success in software SMEs. This is important because to date, is has been reported that software SMEs can have a low process priority. Evidence of the sort presented in this study will help to convince senior managers in software SMEs of the benefits of SPI, which in turn should give rise to increased process prioritisation resulting in greater business success.

Secondly, the research findings have highlighted that much of the SPI that is implemented in software SMEs is of a minor or moderate nature, with over one third of all SPI being reported to be minor in nature. This is representative of a tweaking of the process, rather than implementing wholesales process changes. It is particularly interesting to discover that some of this minor SPI is tacit in nature, with participants reporting that through experience they have extended their mental checklists with respect to the software development process. A good example of this relates to the consideration of long term support and maintenance costs, with several participants reporting that they are more cognisant of this, even at the requirements capture stage.

Although the participants are more aware of the lifetime cost of support and maintenance, there are no records and no evidence of this improvement. Therefore, it is likely that traditional process assessments would not recognise this as a process improvement. However, it is an important improvement in terms of the effectiveness of the software process in supporting the business.

The finding that some SPI is informal and tacit in nature resonates with some of the concepts of agile software development (Fowler and Highsmith 2001), where human capital is entrusted above formal process definition. Whether or not one is a protagonist of agile software development, as a community we must recognise that there are process improvements that are occurring at a human level that would appear to be beyond the scope of traditional process assessments and audits. As outlined in section 9.4, this study has established evidence that informal and tacit process improvement is taking place on a regular basis in software SMEs – important evidence that the research and practicing communities need to integrate into their thinking, and possibly also into future evolutions of their frameworks and models.

The third area in which this research has made an important contribution relates to the influence of situational context in the definition and maintenance of software processes. Many earlier research efforts have commented that situational context is an important consideration for the software process, but no earlier published research has attempted to develop a comprehensive framework of the situational factors affecting the software development process. Therefore, this research study constructed a new reference framework of the situational factors affecting the software development process, drawing factors and wisdom from a from a range of important related contributions, and merging these factors using the constant comparison and memoing techniques from grounded theory (Glaser and Strauss 1976).

Using this new situational factors reference framework, the study investigated the degree of situational change that is evident in software SMEs, with some of the evidence suggesting that situational change may be an important consideration in the interplay between SPI activity and business success. This particular finding emphasises the importance of dynamic capability with respect to the software development process, and according to the evolutionary theory of the firm (Nelson and Winter 1982), companies with greater dynamic capability are likely to be more successful. Dynamic capability is an evolutionary type characteristic and perhaps the software development community as a whole can benefit greatly from increased and structured application of these concepts.

The evidence from this study recommends dynamic capability as an important concept that should be given greater focus in SPI. Process capability alone – either formalised or not – will not propel a business to continual success in the long run. As is the case in the natural world, the key to survival and success is a dynamic capability to continually and appropriately adapt to the changing environment. As Harvey Fineberg, former Provost of Harvard University has stated: "*Evolution doesn't necessarily favour the biggest or the strongest or the fastest and not even the smartest. Evolution favours those creatures best adapted to their environment – that is the sole test of survival and success*" (Fineberg 2011). The findings of this research support the important role of software process evolution in software SMEs and could herald a new movement in SPI, one that emphasises dynamic capability. In this respect, this thesis recommends that the research community should examine the use of dynamic capability concepts in future research endeavours.

A fourth area where this research has made a significant contribution to the body of knowledge relates to our understanding of business objectives in software SMEs. This is the first study to systematically examine business objectives across a number of software SMEs. Using the HSC (Sureshchandar and Leisten 2005) as a comprehensive reference framework for the dimensions of business success in software development organisations, this research has developed a clear understanding of the high and low priority business objectives for software SMEs. This information is important as businesses exist so as to achieve their objectives, and the business processes (including the software development process) should be designed and maintained so as to support the business in achieving its goals. Therefore, it is essential that we have a mechanism for determining business objectives. This study has designed a two-phased engagement for examining business success in software companies, a robust approach that first determines the objectives for the forthcoming period and at a later date checks the degree to which the objectives were achieved. This is the first time that such an approach has been applied in SPI research, and it is believed to offer a reliable mechanism for examining business success in future SPI studies.

10.5.2 Secondary contributions

In addition to the primary contributions identified above, a number of secondary contributions can also be identified. Firstly, the study adopted ISO/IEC 12207 (ISO/IEC 2008) as the software process reference framework, with the participating organisations reporting that ISO/IEC 12207 was considered to be comprehensive in scope. However, the discharge of the SPI activity survey instrument also highlighted a number of limitations in ISO/IEC 12207 in terms of its applicability to software SMEs.

A number of the participating organisations pointed to inadequacies in both the language and the mechanisms adopted in ISO/IEC 12207, suggesting that ISO/IEC 12207 is overly formalised for software SMEs. Perhaps this is inevitable with a large international standard that is designed to address the needs of organisations that can be very large in size. However, ISO/IEC 12207 does offer guidelines on the tailoring of the standard and therefore, the criticism regarding the formalised mechanisms may be misplaced – since the survey instrument incorporated all process areas in ISO/IEC 12207 (so as to ensure the broadest possible scope of inquiry). The discharge of the survey instrument also indicated that there are some shortcomings in the language in ISO/IEC 12207, with some contemporary and commonly used language absent in the standard, for example *refactoring* and *timeboxing.*

Another secondary contribution relates to the profile of SPI that is being implemented in software SMEs in practice. In this respect, the findings of this study extend some of the findings from earlier studies. For example, this study finds that improvements to the requirements management and estimation processes are occurring regularly in software SMEs – confirming the findings from earlier studies. The study also establishes new evidence that software SMEs mix-and-match different software development approaches in order to best suit their environment, hence confirming that there is no one size fits all when it comes to software development. Additional new insights suggest that software SMEs regularly improve the software installation process, as well as the process for tendering and bidding for new business.

The study also finds that there is some commonality in the reported situational changes. This is the first study to attempt to investigate the degree of situational change in software SMEs and the findings suggest that the situational context is in flux. In particular, software SMEs experience rapid rates of growth and decline in headcount, while at the same time experiencing ever-increasing demands for product performance. At present, the prevailing economic climate is challenging and this may be a further factor that has affected the performance of the participating companies. In addition, software SMEs have to navigate the ever-changing world of technology, integrating appropriate new technologies (including compilers, tooling, and operating systems) as they arrive onto the technology landscape. These findings highlight the significant challenges faced by software SMEs, which are so great in magnitude that they may partly explain why software SMEs struggle to focus more energy on SPI matters.

This research has also identified new knowledge concerning the achievement of business objectives in software SMEs. Smaller software development companies appear to be quite adept at achieving business objectives in the areas of aids and subsidies from government, extensions to product offerings and existing clients presenting for new business (refer to Figure 29). However, software SMEs are not as successful when addressing revenue and profit targets, and less successful again regarding new client acquisition targets. These latter observations highlight the volatility of software SMEs.

10.6 Impact on the field

The contributions highlighted above have a number of important impacts for both researchers and practitioners in the field of software processes and SPI. For practitioners, perhaps the most significant outcome of the study is that it appears that there are likely to be benefits from focusing on changes in the situational context. The ability to adapt the process vis-à-vis the degree and nature of the situational change may represent an important capability for organisations. This particular characteristic is described as a dynamic capability in the evolutionary theory of the firm (Nelson and Winter 1982) and firms that possess greater dynamic capabilities are more likely to achieve greater success. Therefore, the central questions for software practitioners should not be: *Which process model should we adopt?* or *Which agile methodology should we use?* Rather, the central questions to ask are: *What type of software development process does my situational context demand?* and *What changes are occurring in my situational context and how do these affect my software development process?* Furthermore, the aspects of the software development setting that should be considered are quite broad, encompassing the profile of the applications under development, the constitution and characteristics of both the management and non-management personnel, the nature of the software requirements, the technology employed in application development, the need to satisfy operational constraints, and the general organisational and business needs. Therefore, a broad range of perspectives should be considered when implementing processes and SPI. By focusing SPI through the lens of situational change, organisations can be more confident that they are making the most effective process improvement investments.

For researchers, there are also some implications regarding to the need to adapt software development processes via SPI. Based on the findings derived in this research, it seems that many of our existing process models and frameworks may not adequately address the need for adaptation. Many of the agile software development approaches such as scrum (Schwaber 1995, Schwaber and Beedle 2002), and XP (Beck 1999), provide a framework for responding to changing requirements. However, agile approaches are prescribed, with little scope to change the process. Furthermore, although agile approaches address the demands of changing requirements, they completely fail to address other aspects of change in software development organisations. For example, if a company moves from having an engaging client that is prepared to pay on a cost reimbursement basis for software product development to having clients that are not able to engage with developers and who insist on fixed price contracts, it may be impractical to continue developing under an agile development methodology.

Process maturity frameworks such as ISO 15504 (ISO/IEC 2004) and CMMI (SEI 2006) are also limited in their ability to adapt the software development process to changing circumstances. Although the most mature stage of process implementation involves process optimisation, very few organisations ever reach this highest level of process implementation and can therefore lose out on the benefits of process adaptation. Software quality standards such as ISO 9000 (ISO 2004) are also deficient in terms of having a robust process improvement mechanism. Therefore, the broader community would benefit from introducing the dynamic capability concept into all existing approaches to software development and SPI. Perhaps a convergence of all of these different approaches, a type of roadmap for software development, could be developed that embraces the situational context as a key consideration, using context and changes to context as the principle driver for software process and SPI decisions.

This study was not designed to examine the nature of SPI in software SMEs but rather to investigate the amount of SPI in software SMEs (and the processes undergoing SPI). However, the data collected identifies that SPI in practice can range from small informal instances of SPI to larger, more formal SPI initiatives. Though not explicitly quantified, the tacit and informal SPI that was detected in this study also has impacts for the research community. In recent years, there has been a gradual increase in the awareness of the human aspects of software development. For example, agile software methods promote individuals over formal processes – and certainly, this is to some extent a very welcome development. However, software development is an intensely human-centric activity and therefore, is very much exposed to the frailties of humans. Therefore, some degree of regulation is required in order to balance the needs of the organisation with the needs of the individuals. Whatever the outcome of future developments in the area of human-centricity integration in software development approaches, it appears that tacit and informal SPI are very much at play today and therefore worthy of much greater attention.

In the software SME domain in particular, there are also impacts for the research community. Although many in the research community appreciate the benefits of software processes and SPI for organisations of all sizes, evidence suggests that in practice, some of these benefits are not realised. In software SMEs, there appears to be a disjoint between the acknowledged theoretical benefits of SPI and the actual practicing of SPI. This is partly owing to the absence of hard evidence in support of the benefits of SPI in software SMEs. While the findings of this research offer strong support for the business benefits of SPI in software SMEs, the future accumulation of additional evidence will likely be required in order to fully convince all members of the community of the benefits of sustaining a strong process focus.

10.7 Research limitations

In conducting this research, every effort has been taken to ensure that the reference frameworks for the investigation are comprehensive and fit-for-purpose. In the case of the SPI activity examination, ISO/IEC 12207 was the chosen reference framework. With over 400 individual process tasks, ISO/IEC 12207 is among the most comprehensive process reference frameworks available at present. For the investigation of situational change, a new framework of the situational factors affecting the software development process was developed as part of this research. This new reference framework contains 170 sub-factors affecting the software development process and is presently the most comprehensive framework of its type. The HSC, the chosen reference framework for the examination of business success in the participating organisations, is the only business performance measurement framework that is dedicated to software development organisations. Furthermore, the HSC is an extension of the most widely used general business performance framework, the BSC (Kaplan and Norton 1992).

Although the chosen reference frameworks for the study are large, comprehensive and leading edge in their respective fields (and the participant feedback indicates that the investigations were comprehensive in nature), no framework can claim to be absolutely complete. Therefore, the study is limited in that there is a possibility that some aspects of the software development process, the situational factors affecting the software development process, and the dimensions of business success could have been overlooked. In particular, in relation to ISO/IEC 12207, it is important to highlight that there may be dimensions of the software development process that are beyond the strict confines of the process reference standard. For example, although there is some direction on the mapping between agile software development and ISO/IEC 12207 (Pikkarainen 2006), certain agile software development practices may not neatly align with the processes in ISO/IEC 12207. However, no single software process reference framework is more comprehensive than ISO/IEC 12207 and it seems unlikely that any significant aspects have been overlooked by the study.

While considerable care was exercised to consistently and accurately elicit the data from the participating organisations, a possible threat to validity concerns the subjectivity of the individual participants when responding to the queries. This particular concern is present in any study that inquires as to the view of a participant regarding some phenomenon. To minimise the impact of this concern, each survey instrument question was carefully drafted and consistently discharged to all participants. However, even these steps cannot completely eliminate the risk of subjective interpretation on the part of participants. A further limitation in this research relates to the new metrics that were established to investigate SPI activity, situational change and business success. Since these metrics have been utilised in this research for the first time, we cannot be certain of their efficacy. Therefore, future utilisation of the data collection techniques and metrics introduced in this study would aid our understanding of the general utility of these new approaches.

This research is also limited in terms of the number of participating organisations. While the study involved accessing a variety of busy personnel in fifteen participating SMEs – and such access is very difficult to realise – the sample size is not especially large from a numerical point of view. However, a considerable depth of information has been gathered and the sample size is large in view of the workload involved in discharging a study of this type. Furthermore, feedback from independent and expert peer-review suggests that the number of participating organisations is certainly sufficient in terms of being indicative of the more general reality. In part, this study acts as a pioneer for increased levels of situational awareness (and process adaptation) in software development, and the associated empirical evidence offers important indications of the utility of these concepts. However, a similar study incorporating a larger number of participating companies could expect to make stronger claims in relation to the generalisability of findings – and perhaps this is an area worth considering for future research. In particular, a larger sample size would be beneficial in assuaging concerns in relation to confounding factors (e.g. an individual company could be successful at a particular time because of many factors, such as timing and marketing, even if the software process was deficient).

10.8 Further research

In the initial stage, this study was conceptually focused on the collection of empirical evidence regarding the nature of the relationship between SPI activity and business success in software SME. However, as the study design unfolded and the concept developed more substance, it became apparent that an examination of SPI activity and business success would be incomplete without a parallel investigation of the degree of situational change. To the surprise of the researcher and despite the frequent references to the importance of situational context in the literature, no general reference framework of the situational factors affecting the software development process had earlier been developed. As a result, this study systematically developed an initial reference framework of the situational factors affecting the software development process.

Although the situational factors reference framework developed as part of this research is broader in scope and depth of information than any existing related reference framework, this general area warrants much additional future research. In particular, the initial framework lacks consensual agreement and this affects not just the constituent factors, but also the labelling and classification scheme. Therefore, important future research should focus on extending the framework and getting general agreement on the contents, language and structure.

Further important future research should focus on identifying the associations between situational factors and software development processes. The development of a model of the relationship between characteristics of software development settings and implementations of software development processes would represent a hugely valuable contribution to the field of software development. Although the development of such a resource would represent a very large undertaking, it is the view of this researcher that eventually our field must make greater strides to incorporate the important role of situational context into software process and SPI decisions. The absence of a resource linking software development settings with software development processes consigns us to a future where the protagonists of various prescribed software development approaches evangelise the absolute authority of their particular approach, only to be eventually swept aside by a new fashion. Software development does not benefit from being subjected to such fashions. There is no one size or style that fits all when it comes to software development processes. The requirements of the process are determined by a complex set of situational circumstances that may well be unique to each organisation, with the circumstances themselves being in constant flux. Therefore, the general domain of situational factors affecting the software development process is viewed as being strategically important to the future of software development.

In addition to researching the role and scope of situational circumstances, much additional evidence of the benefits of a strong software development process focus would greatly incentivise software development companies to invest in SPI. This researcher has witnessed first-hand the struggle to convince software development organisations of the benefits of software processes and SPI. However, if compelling evidence of the benefits of SPI was to be accumulated and distributed to practitioners, then the commitment to SPI in practice would be greatly increased. Therefore, it is the view of this researcher that there is much value to be obtained through future empirical studies into the benefits of software processes and SPI.

In this research, the focus was on the relationship between SPI activity, situational change and business success. However, quality improvements that are the result of SPI activity would represent an interesting and valuable fourth dimension to the inquiry. In this respect, it is recommended that any future related research effort attempts to also include software quality as an additional dimension of inquiry.

Finally, it should be stressed that a software development business can be successful for a host of reasons, not all of which are related to SPI. For example, the business can be successful because of the talent of its individuals, its commitment to quality and general business accountability (McConnell 2011). These particular considerations are not directly related to SPI – though they could result in SPI activity. Therefore, the talent of individuals or the commitment of the employees to quality could be the primary drivers for business success rather than the distinct activity of SPI that we have examined in this research. Future valuable research could examine the influence of these primary drivers on SPI activity.

References

Aaen, I., Borjesson, A. and Mathiassen, L. 2007. SPI agility: How to navigate improvement projects. *Software Process: Improvement and Practice*. 12 (3), pp267-281.

Abrahamsson, P., Salo, O., Ronkainen, J. 2002. *Agile Software Development Methods - Review and Analysis.* VTT Publications Number 478. Finland: VTT Technical Research Centre of Finland.

ADB 2007. *Balanced Scorecard for State-Owned Enterprises.* Technical Assistance Project Team 3933-PRC. Philippines: Asian Development Bank.

Adler, P. S. 2005. The Evolving Object of Software Development. *Organization*. 12 (3), pp401-435.

Aguayo, R. 1991. *Dr. Deming: The American Who Taught the Japanese About Quality.* New York, USA: Fireside Book / Simon and Schuster.

Albrecht, A.J. 1979. Measuring application development productivity. *IN: Proceedings of the IBM Applications Development Symposium,* 14-17 October. IBM Corporation: GUIDE International and SHARE, Inc.

Albrecht, A. J. and Gaffney, J. E. 1983. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering*. 9 (6), pp639-648.

Alqatawna, J., Siddiqi, J., Akhgar, B. and Btoush, M. H. 2009. E-Business Security: Methodological Considerations. *International Journal of Business, Economics, Finance and Management Sciences*. 1 (1), pp47-54.

Anacleto, A., Von Wangenheim, C.G., Salviano, C.F. and Savi, R. 2004. Experiences Gained from Applying ISO/IEC 15504 to Small Software Companies in Brazil. *IN: 4th International Conference on Process Assessment and Improvement,* 27-29 April. Buckinghamshire, U.K.: SPICE User Group.

Andersen, H., Cobbold, I. and Lawrie, G. 2001. Balanced Scorecard implementation in SMEs: reflection in literature and practice. *IN: Proceedings of the fourth SMESME Conference,* May 14-16. 2GC Limited. Aalborg University, 9220 Aalborg, Denmark: Department of Production.

Ansoff, H.I. 1965. *Corporate Strategy.* New York, USA: McGraw-Hill.

Appari, A. and Benaroch, M. 2010. Monetary pricing of software development risks: A method and empirical illustration. *Journal of Systems and Software*. 83 (11), pp2098-2107.

Arthur, L.J. 1993. *Improving Software Quality: An Insider's Guide to TQM.* New York, USA: John Wiley & Sons.

Babbie, E.R. 2009. *The practice of social research.* 12th Edition. Belmont, CA, USA: Wadsworth Publishing Company.

Baddoo, N. and Hall, T. 2003. De-motivators for software process improvement: an analysis of practitioners' views. *Journal of Systems and Software*. 66 (1), pp23-33.

Barafort, B., Di Renzo, B. and Merlan, O. 2002. Benefits Resulting from the Combined Use of ISO/IEC 15504 with the Information Technology Infrastructure Library (ITIL). *IN: Proceedings of the 4th International Conference on Product Focused Software Process Improvement,* 9-11 December. LNCS 2559/2002. Heidelberg / Berlin, Germany: Springer-Verlag.

Barki, H., Rivard, S. and Talbot, J. 2001. An Integrative Contingency Model of Software Project Risk Management. *Journal of Management Information Systems*. 17 (4), pp37-69.

Barnes, D. and Hinton, M. 2008. *The Benefits of e-Business Performance Measurement Systems.* Oxford, UK: CIMA Publishing.

Baskerville, R. and Stage, J.2001. Accommodating emergent work practices: Ethnographic choice of method fragments. *IN:* Fitzgerald, B., Russo, N. and DeGross, J. *Realigning Research and Practice in IS Development: The Social and Organisational Perspective.* New York, NY, USA: Kluwer Academic Publishers. pp12-28.

Basri, S. and O'Connor, R.V. 2011. A Study of Software Development Team Dynamics in SPI. *IN: Proceedings of the 17th European Conference on Systems, Software and Service Process Improvement (EuroSPI 2010),* CCIS 172/2010. Berlin / Heidelberg, Germany: Springer Verlag.

Beck, K. 1999. *Extreme programming explained: embrace change.* Reading, Massachusetts, USA: Addison-Wesley.

Bekkers, W., van de Weerd, I., Brinkkemper, S. and Mahieu, A. 2008. The Influence of Situational Factors in Software Product Management: An Empirical Study. *IN: Proceedings of the Second International Workshop on Software Product Management (IWSPM '08),* Los Alamitos, CA, USA: IEEE Computer Society.

Benaroch, M. and Appari, A. 2010. Financial Pricing of Software Development Risk Factors. *IEEE Software*. 27 (5), pp65-73.

Benediktsson, O., Dalcher, D. and Thorbergsson, H. 2006. Comparison of software development life cycles: a multiproject experiment. *IEE Proceedings - Software*. 153 (3), pp87-101.

Best, M.H. 1990. *The New Competition.* Cambridge, UK: Polity Press.

Bharadwaj, S. G., Varadarajan, P. R. and Fahy, J. 1993. Sustainable Competitive Advantage in Service Industries: A Conceptual Model and Research Propositions. *The Journal of Marketing*. 57 (4), pp83-99.

Biro, M., Ivanyos, J. and Messnarz, R. 2000. Pioneering Process Improvement Experiment in Hungary. *Software Process: Improvement and Practice*. 5 (4), pp213-229.

Blaxter, L., Hughes, C. and Tight, M. 2001. *How to Research.* 2nd Edition. Buckingham, U.K.: Open University Press.

Blind, K. and Hipp, C. 2003. The role of quality standards in innovative service companies: An empirical analysis for Germany. *Technological Forecasting and Social Change*. 70 (7), pp653-669.

Boehm, B. 2002. Get ready for agile methods, with care. *IEEE Computer*. 35 (1), pp64-69.

Boehm, B. 1991. Software Risk Management: Principles and Practices. *IEEE Software*. 8 (1), pp32-41.

Boehm, B. 1988. A spiral model of software development and enhancement. *IEEE Computer*. 21 (5), pp61-72.

Boehm, B. 1981. *Software Engineering Economics.* Upper Saddle River, NJ, USA: Prentice Hall PTR.

Boehm, B., Clark, B., Horowitz, E. 2000. *Software Cost Estimation with Cocomo II.* Upper Saddle River, NJ, USA: Prentice Hall PTR.

Boehm, B. and Turner, R. 2003. *Balancing Agility and Discipline - A Guide for the Perplexed.* Boston, Massachusetts, USA: Pearson Education Limited.

Bourne, M., Mills, J., Wilcox, M., Neely, A. and Platts, K. 2000. Designing, implementing and updating performance measurement systems. *International Journal of Operations & Production Management*. 20 (7), pp754-771.

Bowers, A. N., Sangwan, R. S. and Neill, C. J. 2007. Adoption of XP practices in the industry - A survey. *Software Process: Improvement and Practice*. 12 (3), pp283-294.

Brewer, A. and Hunter, J. 2005. *Foundations of multimethod research.* Thousand Oaks, CA, USA: SAGE.

Brewer, J. and Hunter, A. 1989. *Multimethod research: A synthesis of styles.* Newbury, CA, USA: SAGE.

Brooking, A. 1997. The management of intellectual capital. *Long range planning*. 30 (3), pp364-365.

Brown, M.G. 1996. *Keeping Score: Using the Right Metrics to Drive World-Class Performance.* New York, USA: Quality Resources.

Bryant, A. and Charmaz, K. 2007. *The SAGE Handbook of Grounded Theory.* Thousand Oaks, CA, USA: SAGE.

Bryman, A. 2007. Barriers to Integrating Quantitative and Qualitative Research. *Journal of Mixed Methods Research*. 1 (1), pp8-22.

Bryman, A. 2006. Integrating quantitative and qualitative research: how is it done? *Qualitative Research*. 6 (1), pp97-113.

Bryman, A. 2004. *Social Research Methods.* 2nd Edition. Oxford, UK: Oxford University Press.

Bryman, A. and Bell, E. 2003. *Business Research Methods.* Oxford, UK: Oxford University Press.

Bulmer, M. 1988. *Some reflections upon research in organization.* In: A. Bryman (Ed.) Doing research in organizations, pp. 151-161. London, U.K.: Routledge.

Cameron, J. 2002. Configurable development processes. *Communications of the ACM*. 45 (3), pp72-77.

Casher, J. 1984. How to control project risk and effectively reduce the chance of failure. *Management review*. 73 (6), pp50-54.

Cater-Steel, A. 2002. Software process evaluation: experience report. *Electronic Journal of Information Systems Evaluation*. 5 (2), pp1-13.

Cater-Steel, A. 2001. Process improvement in four small software companies. *IN: 13th Australian Software Engineering Conference (ASWEC 2001),* 26-28 August. Los Alamitos, CA, USA: IEEE Computer Society.

Cater-Steel, A. and Rout, T.2008. SPI Long-Term Benefits: Case Studies of Five Small Firms. *IN:* Oktaba, H. *Software Process Improvement for Small and Medium Enterprises - Techniques and Case Studies.* Hershey, PA, USA: IGI Global.

Cepeda, S., Garcia, S. and Langhout, J. 2008. Is CMMI Useful and Usable in Small Settings? *CrossTalk, The Journal of Defense Software Engineering*. 21 (2), pp14–18.

Chandler, A. D. 1992. Organizational Capabilities and the Economic History of the Industrial Enterprise. *The Journal of Economic Perspectives*. 6 (3), pp79-100.

Chen, J., Zhu, Z. and Xie, H. Y. 2004. Measuring Intellectual Capital: a new model and empirical study. *Journal of Intellectual Capital*. 5 (1), pp195-212.

Chow, C. W., Haddad, K. M. and Williamson, J. E. 1997. Applying the balanced scorecard to small companies. *Management Accounting*. 79 (2), pp21-27.

Chua, W. F. 1986. Radical Developments in Accounting Thought. *Accounting Review*. 61 (4), pp601-633.

Cignoni, G.A. 1999. Rapid Software Process Assessment to Promote Innovation in SMEs. *IN: Proceedings of the European Software Day at the 25th EUROMICRO*

Conference 1999, 10 September. Vienna, Austria: Osterreichische Computer Gesellschaft.

Clarke, P. and O'Connor, R. 2010. Harnessing ISO/IEC 12207 to examine the extent of SPI activity in an organisation. *IN: Proceedings of the 17th Conference on European Systems & Software Process Improvement and Innovation (EuroSPI2010),* 1-3 September. CCIS 99/2010. Heidelberg / Berlin, Germany: Springer-Verlag.

Clarke, P. and O'Connor, R. V. 2012. The situational factors that affect the software development process: Towards a comprehensive reference framework. *Journal of Information and Software Technology*. 54 (5), pp433-447.

Clifford, S. 2005. So many standards to follow, so little pay off [Online]. Available from: <http://www.inc.com/magazine/20050501/management.html> [Accessed 26 April 2012].

Coallier, F. 1994. How ISO 9001 fits into the software world. *IEEE Software*. 11 (1), pp98-100.

Cockburn, A. 2002. *Agile Software Development.* Reading, Massachusetts, USA: Addison-Wesley.

Cockburn, A. and Highsmith, J. 2001. Agile software development, the people factor. *IEEE Computer*. 34 (11), pp131-133.

Cohendet, P., Kern, F., Mehmanpazir, B. and Munier, F. 1999. Knowledge coordination, competence creation and integrated networks in globalised firms. *Cambridge Journal of Economics*. 23 (2), pp225-241.

Coleman, G. and O'Connor, R. 2008. Investigating software process in practice: A grounded theory perspective. *Journal of Systems and Software*. 81 (5), pp772-784.

Colwell and Gillett.1982. Spearman versus Kendall *The Mathematical Gazette* [Online]. 66 (438), pp307-309. [Accessed July 2012].

Constantine, L. 2001. Methodological Agility [Online]. Available from: <http://www.ddj.com/architect/184414743> [Accessed 26 April 2012].

Cooper, D.R. and Emory, C.W. 1995. *Business Research Methods.* 5th Edition. Boston, Massachusetts, USA: Irwin McGraw-Hill.

Corbin, J. and Strauss, A. 2008. *Basics of Qualitative Research.* Thousand Oaks, CA, USA: Sage Publications Limited.

Creswell, J.W. 2003. *Research design: Qualitative, quantitative, and mixed methods approaches.* 2nd Edition. Thousand Oaks, CA, USA: SAGE.

Creswell, J.W. and Plano Clark, V.L. 2007. *Designing and conducting mixed methods research.* Thousand Oaks, CA, USA: SAGE.

Creswell, J.W., Plano Clark, V.L., Gutmann, M.L. 2003. *Advanced Mixed Methods Research Designs.* In: (Tashakkori and Teddlie Eds.) Handbook of Mixed Methods in Social and Behavioral Research. Thousand Oaks, CA, USA: SAGE.

Crone, M. 2002. *A profile of the Irish Software Industry*. Belfast, UK: Northern Ireland Economic Research Center.

Crosby, P.B. 1979. *Quality is Free.* New York, USA: McGraw-Hill.

Cugola, G. 1998. Tolerating deviations in process support systems via flexible enactment of process models. *IEEE Transactions on Software Engineering*. 24 (11), pp982-1001.

Cyert, R. M. and Hedrick, C. L. 1972. Theory of the Firm: Past, Present, and Future; An Interpretation. *Journal of Economic Literature*. 10 (2), pp398-412.

Da Silva, J.G. and Da Cunha, P.R. 2006. Reconciling the Irreconcilable: A Software Development Approach that Combines Agile with Formal. *IN: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS 2006),* 4-7 January. Los Alamitos, California, USA: IEEE Computer Society.

de Cesare, S., Lycett, M., Macredie, R., Patel, C. and Paul, R. 2010. Examining perceptions of agility in software development practice. *Communications of the ACM*. 53 (6), pp126-130.

De Waal, A. A. 2003. Behavioural factors important for the successful implementation and use of performance management systems. *Management Decision*. 41 (8), pp688-699.

Dede and I. Lioufko. 2010. *Situational factors affecting software development process selection.* PhD. Master of ScienceUniversity of Gothenburg:

Deephouse, C., Goldenson, D., Kellner, M. and Mukhopadhyay, T. 1995. The effects of software processes on meeting targets and quality. *IN: Proceedings of the 28th Annual Hawaii International Conference on System Sciences (HICSS 1995),* 3-6 Jan. Los Alamitos, California, USA: IEEE Computer Society.

Delany, S.J. and Cunningham, P. 2008. The Application of Case-Based Reasoning to Early Software Project Cost Estimation and Risk Assessment [Online]. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.9932&rep=rep1&type=pdf> [Accessed 26 April 2010].

DeMarco, T. and Boehm, B. 2002. The agile methods fray. *IEEE Computer*. 35 (6), pp90-92.

Dess, G. G. and Robinson, R. B. 1984. Measuring Organizational Performance in the Absence of Objective Measures: The Case of the Privately-Held Firm and Conglomerate Business Unit. *Strategic Management Journal*. 5 (3), pp265-273.

Dewey, J. 1925. *Experience and nature.* Whitefish, MT, USA: Kessinger.

Dorling, A. 1993. SPICE: Software Process Improvement and Capability dEtermination. *Information and Software Technology*. 35 (6-7), pp404-406.

Douglas, D. 2003. Grounded theories of management: a methodological review. *Management Research News*. 26 (5), pp44-52.

Dyba, T. and Dingsoyr, T. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology*. 50 (9-10), pp833-859.

Dyba, T., Dingsoyr, T. and Moe, N.B. 2004. *Process Improvement In Practice: A handbook for IT Companies.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

Easterby-Smith, M., Thorpe, R. and Jackson, P.R. 2008. *Management Research.* 3rd Edition. Thousand Oaks, CA, USA: SAGE.

Eccles, R. G. 1991. The Performance Measurement Manifesto. *Harvard Business Review*. 69 (1), pp131-137.

El Emam, K. 1998. The Internal Consistency of the ISO/IEC 15504 Software Process Capability Scale. *IN: Proceedings of the 5th IEEE International Symposium on Software Metrics (METRICS '98),* 20-21 March. Los Alamitos, California, USA: IEEE Computer Society.

El Emam, K. and Birk, A. 2000. Validating the ISO/IEC 15504 measures of software development process capability. *Journal of Systems and Software*. 51 (2), pp119-149.

Enterprise Ireland. 2009. *Best Connected Software from Ireland - A strategy for development of the indigenous software industry 2009-2013.* Dublin, Ireland: EI.

Erdogmus, H., Favaro, J. and Strigel, W. 2004. ROI in the Software Industry - Return on Investment - Guest Editors' Introduction. *IEEE Software*. 21 (3), pp18-22.

European Commission. 2003. Commission Recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises. 2003/361/EC. *Official Journal of the European Union*. L (124), pp36-41.

Fayad, M. E., Laitinen, M. and Ward, R. P. 2000. Software Engineering in the Small. *Communications of the ACM*. 43 (3), pp115.

Feiler, P. and Humphrey, W. 1992. *Software Process Development and Enactment: Concepts and Definitions.* CMU/SEI-92-TR-004. Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

Feilzer, M. Y. 2010. Doing Mixed Methods Research Pragmatically: Implications for the Rediscovery of Pragmatism as a Research Paradigm. *Journal of Mixed Methods Research*. 4 (1), pp6-16.

Ferguson, P., Leman, G., Perini, P. 1999. *Software Process Improvement Works!* Pittsburgh, Pennsylvania, USA: CMU/SEI-99-TR-027. Software Engineering Institute, Carnegie Mellon University.

Fernandes, K. J., Raja, V. and Whalley, A. 2006. Lessons from implementing the balanced scorecard in a small and medium size manufacturing organization. *Technovation*. 26 (5-6), pp623-634.

Ferratt, T. and Mai, B. 2010. Tailoring software development. *IN: SIGMIS-CPR '10: Proceedings of the 2010 Special Interest Group on Management Information System's 48th annual conference on Computer personnel research on Computer personnel research,* 20-22 May. New York, NY, USA: ACM.

Ferreira, A.I.F., Santos, G., Cerqueira, R. 2007. Applying ISO 9001:2000, MPS.BR and CMMI to Achieve Software Process Maturity: BL Informatica's Pathway. *IN: Proceedings of the 29th International Conference on Software Engineering (ICSE 2007),* 20-26 May. Los Alamitos, California, USA: IEEE Computer Society.

Fineberg, H. 2011. Are we ready for neo-evolution? [Online]. Available from: <http://www.ted.com/talks/harvey_fineberg_are_we_ready_for_neo_evolution.html> [Accessed 26 April 2012].

Firestone, W. A. 1987. Meaning in Method: The Rhetoric of Quantitative and Qualitative Research. *Educational Researcher*. 16 (7), pp16-21.

Fitzgerald, B., Hartnett, G. and Conboy, K. 2006. Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*. 15 (2), pp200-213.

Fleck, D. 2004. A Process for Very Small Projects. *IN: Proceedings of the 22nd Annual Pacific Northwest Software Quality Conference*, 12-13 October. Portland, Oregon, USA: PNSQC/Pacific Agenda.

Fowler, M., Beck, K., Brant, J. 1999. *Refactoring: improving the design of existing code.* Reading, Massachusetts, USA: Addison-Wesley.

Fowler, M. and Highsmith, J. 2001. The Agile Manifesto. *Software Development.* [Online]. pp28-32. Available from: <http://www.ddj.com/architect/184414755> [Accessed 26 April 2012].

Fuller, A. 2001. *Factors that enable or inhibit Australian SME software development organisations in developing and implementing improved processes.* Brisbane, Australia: Faculty of Communication and Information Technology, Griffith University, Brisbane, Australia.

Galin, D. 2004. *Software Quality Assurance - From theory to implementation.* Harlow, Essex, UK: Pearson Education Limited.

Galliers, R. 1991. *Choosing appropriate information systems research approaches: A revised taxonomy.* In: Galliers, R. (Ed.) Information Systems Research: Issues, Methods and Practical Guidelines. Oxford, UK: Blackwell.

Gautreau, A. and Kleiner, B. H. 2001. Recent trends in performance measurement systems - the balanced scorecard approach. *Management Research News*. 24 (3), pp153-156.

Ghalayini, A. M. and Noble, J. S. 1996. The changing basis of performance measurement. *International Journal of Operations & Production Management*. 16 (8), pp63-80.

Gibson, D., Goldenson, D. and Kost, K. 2006. *Performance results of CMMI-Based Process Improvement.* CMU/SEI-2006-TR-004. Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

Ginsberg, M.P. and Quinn, L.H. 1995. *Process tailoring and the software capability maturity model (CMU/SEI-94-TR-024).* Pittsburgh, PA, USA: Software Engineering Institute, Carnegie Mellon University.

Glaser, B. 1998. *Doing Grounded Theory: Issues and Discussions.* Mill Valley, CA, USA: Sociology Press.

Glaser, B. 1992. *Basics of Grounded Theory Analysis.* California, USA: Sociology Press.

Glaser, B. and Strauss, A. 1976. *The Discovery of Grounded Theory: strategies for qualitative research.* Hawthorne, NY, USA: Aldine de Gruyter.

Glass, R. L. 2001. Agile Versus Traditional: Make love, not war! *Cutter IT Journal*. 14 (12), pp12-18.

Glass, R. L. 1996. Through a glass, darkly: methodologies: bend to fit? *ACM SIGMIS Database*. 27 (1), pp14-16.

Grady, R. 1997. *Successful Software Process Improvement.* Upper Saddle River, New Jersey, USA: Prentice Hall.

Graham, J. 1994. ISO implementation trials and tribulations in small companies. *IN: Northcon/94 Conference Record,* 11-13 October. Piscataway, New Jersey, USA: IEEE.

Grant, R. M. 1996. Prospering in Dynamically-Competitive Environments: Organizational Capability as Knowledge Integration. *Organization Science*. 7 (4), pp375-387.

Greene, J. C., Caracelli, V. J. and Graham, W. F. 1989. Toward a Conceptual Framework for Mixed-Method Evaluation Designs. *Educational Evaluation and Policy Analysis*. 11 (3), pp255-274.

Greer, D. and Conradi, R. 2009. Software project initiation and planning - an empirical study. *IET Software*. 3 (5), pp356-368.

Grimaldi, R. and Torrisi, S. 2001. Codified-tacit and general-specific knowledge in the division of labour among firms: A study of the software industry. *Research Policy*. 30 (9), pp1425-1442.

Haber, A. and Runyon, R. 1980. *General Statistics.* 3rd. Boston, Massachusetts, USA: Addison-Wesley Publishing Company, Inc.

Habra, N., Renault, A., Alexandre, S. and Lopez, M. 2002. OWPL Micro Assessment. *IN: Proceedings of the Workshop on Software Quality at the 24th International Conference on Software Engineering (ICSE 2002),* 19-25 May. Los Alamitos, California, USA: IEEE Computer Society.

Hammersley, M. 1996. *The Relationship between Qualitative and Quantitative Research: Paradigm Loyalty versus Methodological Eclecticism.* In: J.T.E. Richardson (Ed.) Handbook of Research Methods for Psychology and the Social Sciences. Leicester, U.K.: BPS Books.

Hanson, B. 2008. Wither Qualitative/Quantitative? Grounds for Methodological Convergence. *Quality and Quantity*. 42 (1), pp97-111.

Hansson, C., Dittrich, Y., Gustafsson, B. and Zarnak, S. 2006. How agile are industrial software development practices? *Journal of Systems and Software*. 79 (9), pp1295-1311.

Harper, W. 1991. *Statistics.* 6th. 128 Long Acre, London, U.K.: Pitman Publishing.

Harrison, W., Settle, J. and Raffo, D. 2001. Assessing the Value of Improved Predictability due to Process Improvements. *IN: 3rd International Workshop on Economics-Driven Software Engineering Research (EDSER-3),* 14-15 May. Los Alamitos, California, USA: IEEE Computer Society.

Hart, S. 1993. Dimensions of Success in New Product Development: an Exploratory Investigation. *Journal of Marketing Management*. 9 (1), pp23-41.

Harter, D. E. and Slaughter, S. A. 2003. Quality Improvement and Infrastructure Activity Costs in Software Development: A Longitudinal Analysis. *Management Science*. 49 (6), pp784-800.

Haworth, G. O. 1984. Social Work Research, Practice, and Paradigms. *The Social service review*. 58 (3), pp343-357.

Hawrysh, S. P. and Ruprecht, J. 2000. Light Methodologies: It's Like Deja Vu All Over Again. *Cutter IT Journal*. 13 (11), pp4-12.

Hayes, R. H. and Abernathy, W. J. 1980. Managing our way to economic decline. *Harvard Business Review*. 58 (4), pp67-77.

Heejun, P. and Seung, B. 2008. An empirical validation of a neural network model for software effort estimation. *Expert Systems with Applications*. 35 (3), pp929-937.

Heineman, G.T. and Councill, W.T. 2001. *Component-Based Software Engineering: Putting the Pieces Together.* Reading, Massachusetts, USA: Addison-Wesley.

Herbsleb, J., Carleton, A., Rozum, J., Siegel, J. and Zubrow, D. 1994. *Benefits of CMM-Based Software Process Improvement: Initial Results.* Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

Herbsleb, J. and Goldenson, D. 1996. A systematic survey of CMM experience and results. *IN: Proceedings of the 18th International Conference on Software Engineering (ICSE 1996),* 25-29 March. Los Alamitos, California, USA: IEEE Computer Society.

Highsmith, J. 2000. *Adaptive software development: a collaborative approach to managing complex systems.* New York, USA: Dorset House Publishing.

Hinton, P. 1995. *Statistics Explained: A guide for social science students.* 1st. London, United Kingdom: Routledge.

Hollander, E. P. 1974. Processes of Leadership Emergence. *Journal of Contemporary Business*. (Autumn Issue), pp19-33.

Holsti, O.R. 1969. *Content analysis for the social sciences and humanities.* Reading, Massachusetts, USA: Addison-Wesley.

Hudson, M., Smart, A. and Bourne, M. 2001. Theory and practice in SME performance measurement systems. *International Journal of Operations & Production Management*. 21 (8), pp1096-1115.

Humphrey, W.S. 1995. *A Discipline for Software Engineering.* Reading, Massachusetts, USA: Addison-Wesley.

Humphrey, W.S. 1989. *Managing the Software Process.* Reading, Massachusetts, USA: Addison-Wesley.

Hyde, K. and Wilson, D. 2004. Intangible benefits of CMM-based software process improvement. *Software Process: Improvement and Practice*. 9 (4), pp217-228.

IEEE 2004. *Guide to the Software Engineering Book of Knowledge (SWEBOK).* Los Alamitos, CA, USA: IEEE Computer Society.

IEEE 1991. *IEEE Standard Computer Dictionary.* New York, USA: IEEE.

ISO. 2009a. ISO 9000 Essentials [Online]. Available from: <http://www.iso.org/iso/iso_catalogue/management_standards/iso_9000_iso_14000/iso_9000_essentials.htm> [Accessed 26 April 2012].

ISO. 2009b. Management Standards - Benefits [Online]. Available from: <http://www.iso.org/iso/iso_catalogue/management_standards/benefits.htm> [Accessed 26 April 2012].

ISO 2008. *The ISO Survey of Certification 2007.* Geneva, Switzerland: ISO.

ISO 2004. *ISO 9003:2004 - Software Engineering - Guidelines for the application of ISO 9001:2000 to computer software*. Geneva, Switzerland: ISO.

ISO 2000a. *ISO 9000:2000 - Quality management systems - Fundamentals and vocabulary*. Geneva, Switzerland: ISO.

ISO 2000b. *ISO 9001:2000 - Quality Management Systems - Requirements.* Geneva, Switzerland: ISO.

ISO 1994. *ISO 9001 Quality System — Model for Quality Assurance in Design/Development, Production, Installation, and Servicing.* Geneva, Switzerland: ISO.

ISO/IEC 2008. *ISO/IEC 12207-2008 - Systems and software engineering – Software life cycle processes.* Geneva, Switzerland: ISO.

ISO/IEC 2005. *IS0/IEC 15504: Information Technology - Process Assessment, Part 1 to Part 5*. Geneva, Switzerland: International Organisation for Standardization.

ISO/IEC 2004. *ISO/IEC 15504-3 Information technology - Process assessment - Part 3: Guidance on performing an assessment.* Geneva, Switzerland: ISO.

ISO/IEC 2003. *ISO/IEC 15504-2 Information technology - Process assessment - Part 2: Performing an assessment.* Geneva, Switzerland: ISO.

Jacobson D. and Andreosso-O'Callaghan, B. 1996. *Industrial Economics and Organization - A European Perspective.* Maidenhead, Berkshire, United Kingdom: McGraw-Hill.

Jankowicz, A.D. 1995. *Business Research Projects.* 2nd Edition. London, United Kingdom: Thompson Business Press.

Jarvinen, J., Hamann, D. and Van Solingen, R. 1999. On integrating assessment and measurement: towards continuous assessment of software engineering processes. *IN: Proceedings of the 6th International Software Metrics Symposium*, 4-6 November. Los Alamitos, California, USA: IEEE Computer Society.

Jennings, D. F. and Seaman, S. L. 1994. High and low levels of organizational adaptation: An empirical analysis of strategy, structure, and performance. *Strategic Management Journal*. 15 (6), pp459-475.

Jiang, J. J., Klein, G., Hwang, H. G., Huang, J. and Hung, S. Y. 2004. An exploration of the relationship between software development process maturity and project performance. *Information & Management*. 41 (3), pp279-288.

Jick, T. D. 1979. Mixing Qualitative and Quantitative Methods: Triangulation in Action. *Administrative Science Quarterly*. 24 (4, Qualitative Methodology), pp602-611.

Johnson, R. B. and Onwuegbuzie, A. J. 2004. Mixed Methods Research: A Research Paradigm Whose Time Has Come. *Educational Researcher*. 33 (7), pp14-26.

Jones, C. 2008. Development Practices for Small Software Applications. *CrossTalk, The Journal of Defense Software Engineering*. 21 (2), pp9-13.

Jost, A. C. 2008. Small Project Survival Among the CMMI Level 5 Big Processes. *CrossTalk, The Journal of Defense Software Engineering*. 21 (2), pp22-26.

Jung, H. W., Hunter, R., Goldenson, D. R. and El Emam, K. 2001. Findings from Phase 2 of the SPICE trials. *Software Process Improvement and Practice*. 6 (4), pp205-242.

Kaplan, R. S. and Norton, D. P. 1992. The Balanced Scorecard - Measures That Drive Performance. *Harvard Business Review*. 70 (1), pp71-79.

Karlstrom, D. and Runeson, P. 2006. Integrating agile software development into stage-gate managed product development. *Empirical Software Engineering*. 11 (2), pp203-225.

Kautz, K. 1998. Software process improvement in very small enterprises: does it pay off? *Software Process: Improvement and Practice*. 4 (4), pp209-226.

Keane, B. and Richardson, I. 2005. Quality: Attitudes and Experience within the Irish Software Industry. *Lecture Notes in Computer Science: Software Process Improvement (Volume 3792/2005)*. pp49-58.

Keil, M., Cule, P. E., Lyytinen, K. and Schmidt, R. C. 1998. A framework for identifying software project risks. *Communications of the ACM*. 41 (11), pp76-83.

Kennerley, M. and Neely, A. 2002. *Performance Measurement Frameworks: A Review*. IN: Business Performance Measurement - Theory and Practice. Cambridge, UK: Cambridge University Press.

Kettunen, V., Kasurinen, J., Taipale, O. and Smolander, K. 2010. A study on agility and testing processes in software organizations. *IN: ISSTA '10: Proceedings of the 19th international symposium on Software testing and analysis,* June 12-16 2010. New York, NY, USA: ACM.

Khurshid, N., Bannerman, P. and Staples, M. 2009. Overcoming the First Hurdle: Why Organizations Do Not Adopt CMMI. *IN: Proceedings of the International Conference on Software Process (ICSP 2009),* 16-17 May. LNCS 5543/2009. Berlin / Heidelberg, Germany: Springer-Verlag.

Kitchenham, B. 1991. Making process predictions. *IN:* Fenton, N.E. *Software Metrics: A Rigorous Approach.* London, United Kingdom: Chapman & Hall.

Knigge, L. and Cope, M. 2006. Grounded visualization: integrating the analysis of qualitative and quantitative data through grounded theory and visualization. *Environment and Planning*. 38 (11), pp2021-2037.

Koh, T. W., Selamat, M. H. and Ghani, A. 2008. Exponential Effort Estimation Model Using Unadjusted Function Points. *Information Technology Journal*. 7 (6), pp830-839.

Kolodner, J. 1993. *Case-Based Reasoning.* California: Morgan Kaufmann.

Koopmans, L. 1981. *An introduction to contemporary statistics.* 1st. Belmont, CA, USA: PWS Publishers.

Kotey, B. and Meredith, G. G. 1997. Relationships among Owner/Manager Personal Values, Business Strategies, and Enterprise Performance. *Journal of Small Business Management*. 35 (2), pp37.

Kranzler, J. 2003. *Statistics for the Terrified.* 3rd. Upper Saddle River, NJ, USA: Prentice Hall.

Krathwohl, D.R. 2004. *Methods of educational and social science research: an integrated approach.* 2nd Edition. Long Grove, IL, USA: Waveland.

Kruchten, P. and Royce, W. 1996. A rational development process. *Crosstalk, The Journal of Defense Software Engineering*. 9 (7), pp11-16.

Lakshman, M., Sinha, L., Biswas, M., Charles, M. and Arora, N. 2000. Quantitative Vs qualitative research methods. *Indian journal of pediatrics*. 67 (5), pp369-377.

Laporte, C.Y., Desharnais, J.M., Abouelfattah, M., Bamba, J.C., Renault, A. and Habra, N. 2005. Initiating Software Process Improvement in Small Enterprises: Experiments with Micro-Evaluation Framework. *IN: Proceedings of the International Conference on Software Development,* 27 May - 1 June. University of Iceland: University of Iceland Press.

Lawlis, P., Flowe, R. and Thordahl, J. 1995. A Correlational Study of the CMM and Software Development Performance. *Crosstalk, The Journal of Defense Software Engineering*. 8 (9), pp21-25.

Lee, A. S. and Baskerville, R. L. 2003. Generalizing Generalizability in Information Systems Research. *Information Systems Research*. 14 (3), pp221-243.

Lempert, L. 2007. Asking questions of the data: memo writing in the grounded theory tradition. *IN:* Bryant, A. and Charmaz, K. *The SAGE handbook of grounded theory.* Thousand Oaks, CA, USA: Sage. pp245-264.

Lepasaar, M., Varkoi, T. and Jaakkola, H. 2001. Models and Success Factors of Process Change. *IN: Proceedings of the 3rd International Conference on Product Focused Software Process Improvement*, 10-13 September. LNCS 2188/2001. Berlin / Heidelberg, Germany: Springer-Verlag.

Leung, H. K. N. 1999. Slow Change of Information System Development Practice. *Software Quality Journal*. 8 (3), pp197-210.

Leung, H. K. N. and Yuen, T. C. F. 2001. A process framework for small projects. *Software Process: Improvement and Practice*. 6 (2), pp67-83.

Lim, A. H. L. and Lee, C. S. 2008. Integrated model driven business evaluation methodology for strategic planning. *International Journal of Business Information Systems*. 3 (4), pp333-355.

Lindvall, M. and Rus, I. 2000. Process diversity in software development. *IEEE Software*. 17 (4), pp14-18.

Lowstedt, J. and Stjernberg, T. 2006. *Producing Management Knowledge - Research as Practice.* Abindgton, Oxon, U.K.: Routledge.

Lukacs, E. 2005. The Economic Role of SMEs in World Economy, especially in Europe. *European Integration Studies*. 4 (1), pp3-12.

Lycett, M., Macredie, R. D., Patel, C. and Paul, R. J. 2003. Migrating agile methods to standardized development practice. *IEEE Computer*. 36 (6), pp79-85.

Lynch, R.L. and Cross, K.F. 1990. *Measure Up! Yardstick for Continuous Improvement.* Cambridge, Massachusetts, USA: Basil Blackwell.

Lyytinen, K., Mathiassen, L. and Ropponen, J. 1998. Attention shaping and software risk-A categorical analysis of four classical risk management approaches. *Information Systems Research*. 9 (3), pp233-255.

MacCormack, A. and Verganti, R. 2003. Managing the Sources of Uncertainty: Matching Process and Context in Software Development. *Journal of Product Innovation Management*. 20 (3), pp217-232.

Magee, S. and Thiele, D. 2004. Engineering process standards: state of the art and challenges. *IT Professional*. 6 (5), pp38-44.

Maidique, M. A. and Zirger, B. J. 1985. The new product learning cycle. *Research Policy*. 14 (6), pp299-313.

Mathiassen, L., Ngwenyama, O. K. and Aaen, I. 2005. Managing change in software process improvement. *IEEE Software*. 22 (6), pp84-91.

McAdam, R. and Fulton, F. 2002. The impact of the ISO 9000:2000 quality standards in small software firms. *Managing Service Quality*. 12 (5), pp336-345.

McCaffery, F., McFall, D. and Wilkie, F.G. 2005. Improving the Express Process Appraisal Method. *IN: Proceedings of the 6th International Conference on Product Focused Software Process Improvement*, 13-15 June. LNCS 3547/2005. Berlin / Heidelberg, Germany: Springer-Verlag.

McCaffery, F., Taylor, P. S. and Coleman, G. 2007. Adept: A Unified Assessment Method for Small Software Companies. *IEEE Software*. 24 (1), pp24-31.

McConnell, S. 2011. GTAC 2011 Closing Keynote: Secrets of World Class Software Organisations [Online]. Available from: <www.youtube.com/watch?v=EE-t5J7hnHE> [Accessed 26 April 2012].

McConnell, S. 2002. Closing the gap. *IEEE Software*. 19 (1), pp3-5.

McCormack, K. and Johnson, W. 2001. *Business Process Orientation - Gaining the E-Business Competitive Advantage.* Florida, USA: CRC Press.

McKenzie, F. and Shilling, M. 1998. Avoiding performance measurement traps: ensuring effective incentive design and implementation. *Compensation and Benefits Review*. 30 (4), pp57-65.

Merriam, S.B. 1998. *Qualitative Research and Case Study Applications in Education.* 2nd Edition. San Francisco, California, USA: Jossey-Bass Publishers.

Miles, M. and Huberman, A. 1984. *Qualitative Data analysis.* London, United Kingdom: SAGE.

Miller, S. 2006. Mixed Methods As Methodological Innovations: Problems And Prospects. *Methodological Innovations Online*. 1 (1), pp1-7.

Miluk, G. 2005. Results of a field study of CMMI for small settings using rapid applied ethnography. *IN: Proceedings of the 1st International Research Workshop for Process Improvement in Small Settings*, 19-20 October. CMU-SEI-2006-SR-001. Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

Mingers, J. 2001. Combining IS Research Methods: Towards a Pluralist Methodology. *Information Systems Research*. 12 (3), pp240-260.

Mishra, D. and Mishra, A. 2009. Software Process Improvement in SMEs: A Comparative View. *Computer Science and Information Systems (ComSIS)*. 6 (1), pp111-140.

Molina-Castillo, F. and Munuera-Aleman, J. 2009. The joint impact of quality and innovativeness on short-term new product performance. *Industrial Marketing Management*. 38 (8), pp984-993.

Moore, J.W. 1998. IEEE/EIA 12207 As the Foundation for Enterprise Software Processes. *IN: The Joint 1998 Proceedings of the Pacific Northwest Software Quality Conference and the 8th International Conference on Software Quality*, 13-14 October. Portland, Oregon, USA: PNSQC/Pacific Agenda.

Morgan, R. E. and Strong, C. A. 2003. Business performance and dimensions of strategic orientation. *Journal of Business Research*. 56 (3), pp163-176.

Myers, M.D. 2009. *Qualitative Research in Business and Management.* Thousand Oaks, CA, USA: SAGE.

Myers, M. D. 1997. Qualitative research in information systems. *MIS Quarterly*. 21 (2), pp241-242.

Naslund, D. 2002. Logistics needs qualitative research - especially action research. *International Journal of Physical Distribution and Logistics Management*. 32 (5), pp321-338.

Neely, A.D., Adams, C. and Kennerley, M. 2002. *The Performance Prism: the scorecard for measuring and managing business success.* London, United Kingdom: Prentice Hall.

Nelson, R.R. and Winter, S. 1982. *An Evolutionary Theory of Economic Change.* Cambridge, Massachusetts, USA: The Balknap Press of Harvard University Press.

Newman, I. and Benz, C.R. 1998. *Qualitative-quantitative research methodology: exploring the continuum.* Carbondale, IL, USA: University of Illinois Press.

Niazi, M. 2006. Software Process Improvement: A Road to Success. *IN: Proceedings of the 7th International Conference on Product Focused Software Process Improvement,* 12-14 June. LNCS 4034/2006. Berlin / Heidelberg, Germany: Springer.

Niazi, M. and Babar, M. A. 2009. Identifying high perceived value practices of CMMI level 2: An empirical study. *Information and Software Technology*. 51 (8), pp1231-1243.

Niazi, M., Babar, M.A. and Ibrahim, S. 2008. An Empirical Study Identifying High Perceived Value Practices of CMMI Level 2. *IN: Proceedings of the 9th International Conference on Product Focused Software Process Improvement*, 23-25 June. LNCS 5089/2008. Berlin / Heidelberg, Germany: Springer-Verlag.

Nikitina, N. and Kajko-Mattsson, M. 2010. Impact of Growing Business on Software Processes. *IN: Proceedings of the 17th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2010),* CCIS 99/2010. Berlin Heidelberg: Springer.

Nonaka, I. and Toyama, R. 2005. The theory of the knowledge-creating firm: subjectivity, objectivity and synthesis. *Industrial and Corporate Change*. 14 (3), pp419-436.

Oates, B. 2006. *Researching Information Systems and Computing.* London, United Kingdom: Sage Publications Limited.

O'Brien, R. 1998. An Overview of the Methodological Approach of Action Research [Online]. Available from: <http://www.web.ca/~robrien/papers/arfinal.html> [Accessed 26 April 2012].

O'Brien, R. 1979. The Use of Pearson's with Ordinal Data. *American Sociological Review*. 44 (5), pp851-857.

O'Connor, R.V. and Coleman, G. 2007. An Investigation of Barriers to the Adoption of Software Process Best Practice Models. *IN: Proceedings of the 18th Australasian Conference on Information Systems,* 5-7 December. Toowoomba, Australia: The University of Southern Queensland.

Oskarsson, O. and Glass, R.L. 1996. *An ISO 9000 Approach to Building Quality Software.* New Jersey, USA: Prentice Hall.

Osterwalder, A. and Pigneur, Y. 2002. An e-business model ontology for modeling e-business. *IN: 15th Bled Electronic Commerce Conference,* June 17-19.

Pall, G. 1987. *Quality Process Management.* Englewood Cliffs, NJ, USA: Prentice Hall.

Palmer, S.R. and Felsing, J. 2002. *A practical guide to feature-driven development.* Upper Saddle River, New Jersey, USA: Prentice Hall.

Pandey, I. M. 2005. Balanced Scorecard: Myth and Reality. *Vikalpa*. 30 (1), pp51-66.

Parker, C. 2000. Performance Measurement. *Work Study*. 49 (2), pp63-66.

Patton, M.Q. 1990. *Qualitative Evaluation and Research Methods.* 2nd Edition. Newbury, CA, USA: Sage.

Paulk, M.C. 1998. Using the Software CMM in Small Organizations. *IN: The Joint 1998 Proceedings of the Pacific Northwest Software Quality Conference and the 8th International Conference on Software Quality,* 13-14 October. Portland, Oregon, USA: PNSQC/Pacific Agenda.

Paulk, M.C., Curtis, B., Chrissis, M.B. 1993. *Capability Maturity Model for Software.* Version 1.1. CMU/SEI-93-TR-24. Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

Peile, C. and McCouat, M. 1997. The Rise of Relativism: The Future of Theory and Knowledge Development in Social Work. *The British Journal of Social Work*. 27 (3), pp343-360.

Petersen, K. and Wohlin, C. 2010. The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering*. 15 (6), pp654-693.

Petersen, K. and Wohlin, C. 2009. Context in industrial software engineering research. *IN: Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement,* Washington, DC, USA: IEEE Computer Society.

Peterson, D. K., Kim, C., Kim, J. H. and Tamura, T. 2002. The perceptions of information systems designers from the United States, Japan, and Korea on success and failure factors. *International Journal of Information Management*. 22 (6), pp421-439.

Petter, S.C. and Gallivan, M.J. 2004. Toward a Framework for Classifying and Guiding Mixed Method Research in Information Systems. *IN: Proceedings of the 37th Hawaii International Conference on System Sciences,* January 5-8. Los Alamitos, California, USA: IEEE Computer Society.

Pikkarainen, M. 2006. *Mapping Agile Software Development onto ISO 12207.* Published online at www.agile-itea.org: Agile Consortium.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P. and Still, J. 2008. The impact of agile practices on communication in software development. *Empirical Software Engineering*. 13 (3), pp303-337.

Pino, F. J., Garcia, F. and Piattini, M. 2008. Software process improvement in small and medium software enterprises: a systematic review. *Software Quality Journal*. 16 (2), pp237-261.

Pino, F. J., Pardo, C., Garcia, F. and Piattini, M. 2010. Assessment methodology for software process improvement in small organizations. *Information and Software Technology*. 52 (10), pp1044-1061.

Poulin, L. A. 2007. Achieving the Right Balance Between Process Maturity and Performance. *IEEE Canadian Review*. 56 (-), pp23-26.

Punch, M. 1998. *Introduction to Social Research: Quantitative and Qualitative Approaches.* Thousand Oaks, CA, USA: SAGE.

Punch, M. 1986. *The Politics and Ethics of Fieldwork.* Beverly Hills, CA, USA: SAGE.

Putnam, L. 1978. A General Empirical Solution to the Macro Software Sizing and Estimating Problem. *IEEE Transactions on Software Engineering*. 4 (4), pp345-361.

Reid, G. C. and Smith, J. A. 2000. What Makes a New Business Start-Up Successful? *Small Business Economics*. 14 (3), pp165-182.

Reifer, D. J. 2002. How Good Are Agile Methods? *IEEE Software*. 19 (4), pp16-18.

Reilly, J. 1997. *Understanding Statistics.* 1st. Dublin, Ireland: Folens Publishers.

Richardson, I. and von Wangenheim, C. G. 2007. Guest Editors' Introduction: Why are Small Software Organizations Different? *IEEE Software*. 24 (1), pp18-22.

Rico, D.F. 2004. *ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers.* Fort Lauderdale, Florida, USA: J. Ross Publishing.

Ropponen, J. and Lyytinen, K. 2000. Components of software development risk: how to address them? A project manager survey. *IEEE Transactions on Software Engineering*. 26 (2), pp98-112.

Rorty, R. 1999. *Philosophy and social hope.* London, United Kingdom: Penguin Books.

Rossman, G. B. and Wilson, B. L. 1994. Numbers and words revisited: Being "shamelessly eclectic". *Quality & Quantity*. 28 (3), pp315-327.

Rossman, G. B. and Wilson, B. L. 1985. Numbers and Words: Combining Quantitative and Qualitative Methods in a Single Large-Scale Evaluation Study. *Evaluation Review*. 9 (5), pp627-643.

Rout, T. P., El Emam, K., Fusani, M., Goldenson, D. R. and Jung, H. W. 2007. SPICE in retrospect: Developing a standard for process assessment. *Journal of Systems and Software*. 80 (9), pp1483-1493.

Royce, W. 1970. Managing the development of large software systems: concepts and techniques. *IN: Western Electric Show and Convention Technical Papers,* 25-28 August. Los Alamitos, California, USA: IEEE Computer Society.

Rozum, J.A. 1993. *Concepts on Measuring the Benefits of Software Process Improvements.* CMU/SEI-93-TR-009. Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

Ruparelia, N. B. 2010. Software development lifecycle models. *SIGSOFT Software Engineering Notes*. 35 (3), pp8-13.

Ryan, S. and O'Connor, R. V. 2009. Development of a team measure for tacit knowledge in software development teams. *Journal of Systems and Software*. 82 (2), pp229-240.

Saastamoinen, I. and Tukiainen, M. 2004. Software Process Improvement in Small and Medium Sized Software Enterprises in Eastern Finland: A State-of-the-Practice Study. *IN: Proceedings of the 11th European Conference on Software Process Improvements (EuroSPI 2004),* 10-12 November. LNCS 3281/2004. Berlin / Heidelberg, Germany: Springer-Verlag.

Sale, J. E. M., Lohfeld, L. H. and Brazil, K. 2002. Revisiting the Quantitative-Qualitative Debate: Implications for Mixed-Methods Research. *Quality and Quantity*. 36 (1), pp43-53.

Sanders, M. 1998. *The SPIRE Handbook. Better, Faster, Cheaper Software Development in Small Organisations.* DCU, Dublin, Ireland: Centre for Software Engineering Limited.

SCAMPI Upgrade Team 2006. *Appraisal Requirements for CMMI (ARC).* Version 1.2. CMU/SEI-2006-TR-011. Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

Schuler, K. 1995. Preparing for ISO 9000 registration: the role of the technical communicator. *IN: Proceedings of the 13th annual international conference on Systems documentation (SIGDOC '95),* 30 September - 3 October. New York, USA: ACM.

Schwaber, K. 1995. SCRUM Development Process. *IN: Business Object Design and Implementation Workshop at the 10th Annual Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA 1995),* 15-19 October. Berlin / Heidelberg, Germany: Springer-Verlag.

Schwaber, K. and Beedle, M. 2002. *Agile Software Development with SCRUM.* Upper Saddle River, New Jersey, USA: Prentice Hall.

Schwandt, T.A. 2007. *The SAGE Dictionary of Qualitative Inquiry.* 2nd Edition. Thousand Oaks, CA, USA: SAGE.

Scott, L., Carvalho, L., Jeffery, R., D'Ambra, J. and Becker-Kornstaedt, U. 2002. Understanding the use of an electronic process guide. *Information and Software Technology*. 44 (10), pp601-616.

SEI 2009a. *Process Maturity Profile, CMMI for Development, SCAMPI Class A Appraisal Results, 2008 End-Year Update.* Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

SEI. 2009b. SEI Statistics and History [Online]. Available from: <http://www.sei.cmu.edu/about/statisticshistory.cfm> [Accessed 26 April 2012].

SEI 2006. *CMMI for Development, Version 1.2.* CMU/SEI-2006-TR-008. Pittsburgh, PA, USA: Software Engineering Institute.

Serrano, M.A., Montes de Oca, C. and Cedillo, K. 2005. An Experience on Implementing the CMMI in a Small Organisation Using the Team Software Process. *IN: Proceedings of the 1st International Research Workshop for Process Improvement in Small Settings,* 19-20 October. CMU/SEI-2006-SR-001. Pittsburgh, Pennsylvania, USA: Software Engineering Institute, Carnegie Mellon University.

Shewhart, W.A. 1931. *Economic Control of Quality of Manufactured Product.* New York, USA: D. Van Nostrand Company.

Silk, S. 1998. Automating the balanced scorecard. *Management Accounting*. 79 (11), pp38-40.

Skrinjar, R., Bosilj-Vuksic, V. and Indihar-Stemberger, M. 2008. The impact of business process orientation on financial and non-financial performance. *Business Process Management Journal*. 14 (5), pp738-754.

Sommerville, I. 2007. *Software Engineering.* 8th Edition. Harlow, Essex, U.K.: Pearson Education Limited.

Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P. and Murphy, R. 2007. An exploratory study of why organizations do not adopt CMMI. *Journal of Systems and Software*. 80 (6), pp883-895.

Stapleton, J. 1997. *Dynamic systems development method - The method in practice.* Harlow, Essex, UK: Pearson Education Limited.

Stelzer, D., Mellis, W. and Herzwurm, G. 1997. A critical look at ISO 9000 for software quality management. *Software Quality Journal*. 6 (2), pp65-79.

Stelzer, D., Mellis, W. and Herzwurm, G. 1996. Software Process Improvement via ISO 9000? Results of Two Surveys among European Software Houses. *Software Process: Improvement and Practice*. 2 (3), pp197-210.

Stevenson, T. H. and Barnes, F. C. 2001. Fourteen Years of ISO 9000: Impact, Criticisms, Costs, and Benefits. *Business Horizons*. 44 (3), pp45-51.

Subramanian, G. H., Klein, G., Jiang, J. J. and Chan, C. 2009. Balancing four factors in system development projects. *Communications of the ACM*. 52 (10), pp118-121.

Sureshchandar, G. S. and Leisten, R. 2005. Holistic Scorecard: strategic performance measurement and management in the software industry. *Measuring Business Excellence*. 9 (2), pp12-29.

Sutton, S. M. 2000. The role of process in software start-up. *IEEE Software*. 17 (4), pp33-39.

Syed-Abdullah, S., Holcombe, M. and Gheorge, M. 2006. The Impact of an Agile Methodology on the Well Being of Development Teams. *Empirical Software Engineering*. 11 (1), pp143-167.

Tapanya. 2004. *Examining the factors which influence performance measurement and management in the Thai Banking Industry: An application of the balanced scorecard framework.* PhD Thesis. Murdoch University: Perth, Western Australia.

Tashakkori, A. and Creswell, J. W. 2007. Editorial: Exploring the Nature of Research Questions in Mixed Methods Research. *Journal of Mixed Methods Research*. 1 (3), pp207-211.

Tashakkori, A. and Teddlie, C. 2003. *Handbook of Mixed Methods in Social and Behavioral Research.* Thousand Oaks, CA, USA: SAGE.

Tashakkori, A. and Teddlie, C. 1998. *Mixed methodology: Combining qualitative and quantitative approaches.* Applied Social Research Methods Series (Vol. 46). Thousand Oaks, CA, USA: SAGE.

Taylor, A.S. 2010. *Ethnography in Ubiquitous Computing.* In Krumm, J. (Ed.) Ubiquitous Computing Fundamentals. Boca Raton, Florida, USA: Chapman & Hall/CRC.

Teddlie, C. and Tashakkori, A. 2009. *Foundations of mixed methods research.* Thousand Oaks, CA, USA: SAGE.

Terlaak, A. and King, A. A. 2006. The effect of certification with the ISO 9000 Quality Management Standard: A signaling approach. *Journal of Economic Behavior & Organization*. 60 (4), pp579-602.

Tilley, T., Cole, R., Becker, P. 2005. *A Survey of Formal Concept Analysis Support for Software Engineering Activities.* IN: Formal Concept Analysis. LNCS 3626/2005. Berlin / Heidelberg, Germany: Springer-Verlag.

Truex, D., Baskerville, R. and Travis, J. 2000. Amethodical systems development: the deferred meaning of systems development methods. *Accounting, Management and Information Technologies*. 10 (1), pp53-79.

Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M., Cheng, C. K., Permadi, R. B. and Feldt, R. 2012. Evaluation and Measurement of Software Process Improvement: A Systematic Literature Review. *IEEE Transactions on Software Engineering*. 38 (2), pp398-424.

Valtanen, A. and Ahonen, J. 2008. Big Improvements with Small Changes: Improving the Processes of a Small Software Company. *IN: Proceedings of the 9th International Conference on Product Focused Software Process Improvement*, 23-25 June. LNCS 5089/2008. Berlin / Heidelberg, Germany: Springer-Verlag.

Van de Grift, T. 2004. Coupling pair programming and writing: learning about students' perceptions and processes. *ACM SIGCSE Bulletin*. 36 (1), pp2-6.

Van Der Wiele, T. and Brown, A. 1997. ISO 9000 series experiences in small and medium-sized enterprises. *Total Quality Management & Business Excellence*. 8 (2), pp300-304.

Van Solingen, R. 2004. Measuring the ROI of Software Process Improvement. *IEEE Software*. 21 (3), pp32-38.

Van Solingen, R. 2001. The Cost and Benefits of Software Process Improvement. *IN: Proceedings of the 8th European Conference on Information Technology Evaluation,* 17-18 September. Reading, U.K.: MCIL.

Van Vliet, V. 2000. *Software Engineering - Principles and Practice.* Second Edition. Chichester, West Sussex, England: John Wiley & Sons.

Vergidis, K., Tiwari, A. and Majeed, B. 2008. Business Process Analysis and Optimization: Beyond Reengineering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*. 38 (1), pp69-82.

Verschuren, P. J. M. 2003. Case study as a research strategy: some ambiguities and opportunities. *International Journal of Social Research Methodology*. 6 (2), pp121-140.

Visconti, M. and Cook, C.R. 2004. An Ideal Process Model for Agile Methods. *IN: Proceedings of the 5th International Conference on Product Focused Software Process Improvement*, 5-8 April. LNCS 3009/2004. Berlin / Heidelberg, Germany: Springer-Verlag.

Von Wangenheim, C. G., Anacleto, A. and Salviano, C. F. 2006. Helping small companies assess software processes. *IEEE Software*. 23 (1), pp91-98.

Von Wangenheim, C. G., Weber, S., Hauck, J. C. R. and Trentin, G. 2006. Experiences on establishing software processes in small companies. *Information and Software Technology*. 48 (9), pp890-900.

Wallace, L. and Keil, M. 2004. Software project risks and their effect on outcomes. *Communications of the ACM*. 47 (4), pp68-73.

Wang, X., O'Conchuir, E. and Vidgen, R. 2008. A paradoxical perspective on contradictions in agile software development. *IN: Proceedings of the 16th European Conference on Information Systems (ECIS 2008)*, 9-11 June. Berlin / Heidelberg, Germany: Springer-Verlag.

Watson, I. and Marir, F. 1994. Case-Based Reasoning: A Review. *The Knowledge Engineering Review*. 9 (4), pp327-354.

Wegelius, H. and Johansson, M. 2007. Practical Experiences on Using SPICE for SPI in an Insurance Company. *IN: EuroSPI 2007 Industrial Proceedings*, 26-28 September. Potsdam, Germany: ASQF.

Whitworth, E. and Biddle, R. 2007. The Social Nature of Agile Teams. *IN: Proceedings of the AGILE 2007*, 13-17 August. Los Alamitos, California, USA: IEEE Computer Society.

Wilkie, F. G., McFall, D. and McCaffery, F. 2005. An evaluation of CMMI process areas for small- to medium-sized software development organisations. *Software Process: Improvement and Practice*. 10 (2), pp189-201.

Williams, L.A. and Kessler, R.R. 2000. The effects of "pair-pressure" and "pair-learning" on software engineering education. *IN: Proceedings of the 13th Conference on Software Engineering Education & Training*, 6-8 March. Los Alamitos, California, USA: IEEE Computer Society.

Xu, P. and Ramesh, B. 2007. Software Process Tailoring: An Empirical Investigation. *Journal of Management Information Systems*. 24 (2), pp293-328.

Yang, Y. H. 2001. Software quality management and ISO-9000 implementation. *Industrial Management & Data Systems*. 101 (7), pp329-338.

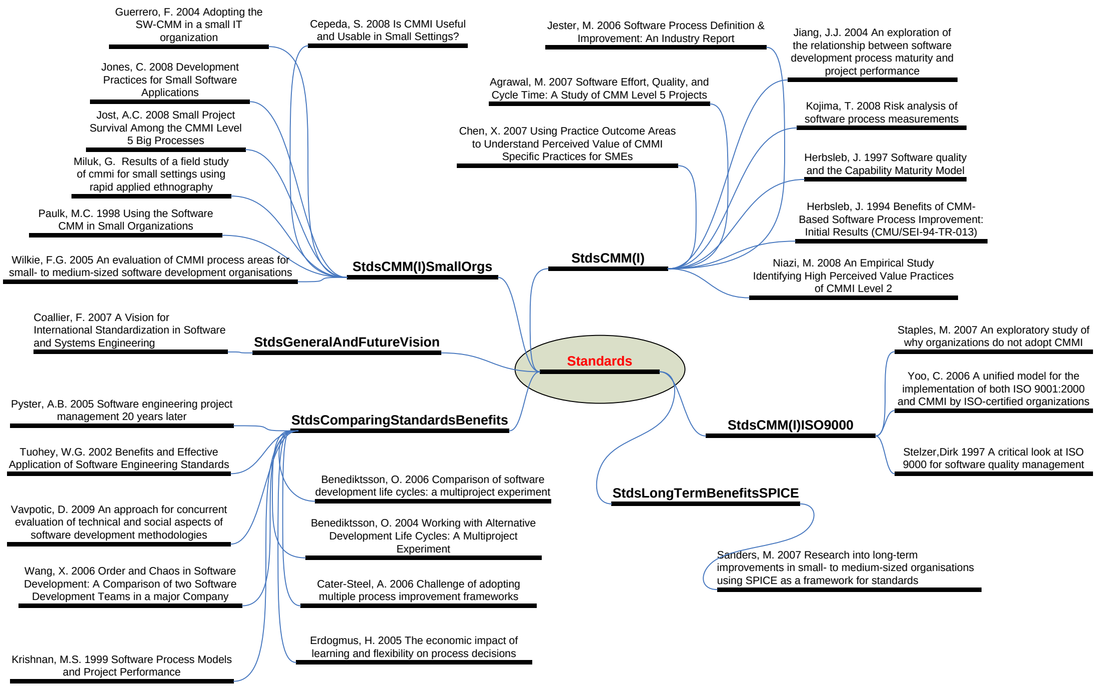
Yin, R.K. 1994. *Case Study Research: Design and Methods*. 2nd Edition. Beverly Hills, CA, USA: SAGE.

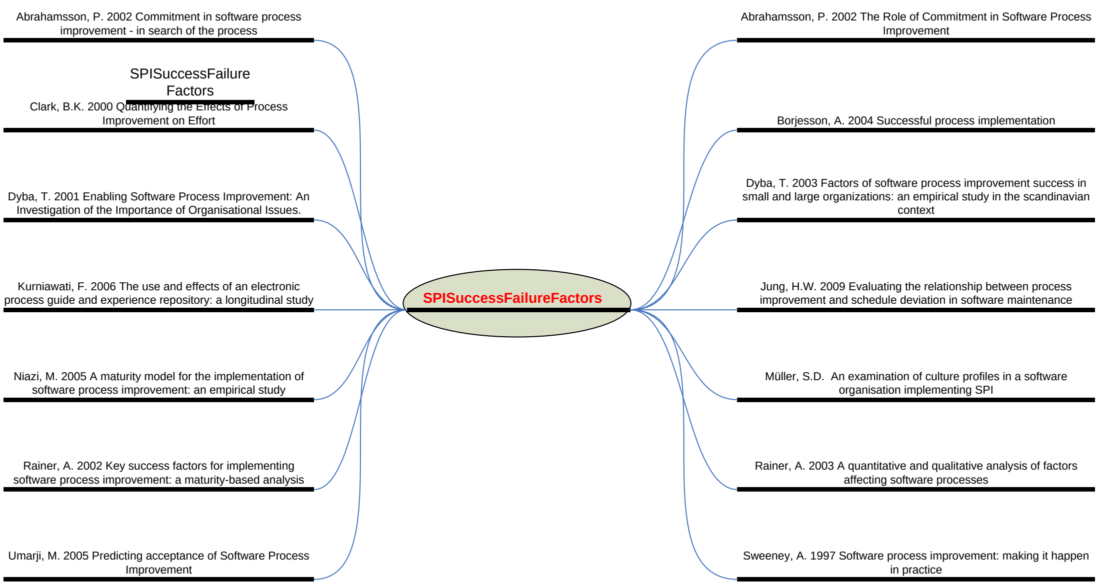
Zahran, S. 2002. Software Process Improvement using the Capability Maturity Model "CMM" [Online]. Available from: <http://www.imamu.edu.sa/topics/IT/IT%206/Software%20Process%20Improvement.pdf> [Accessed 26 April 2012].

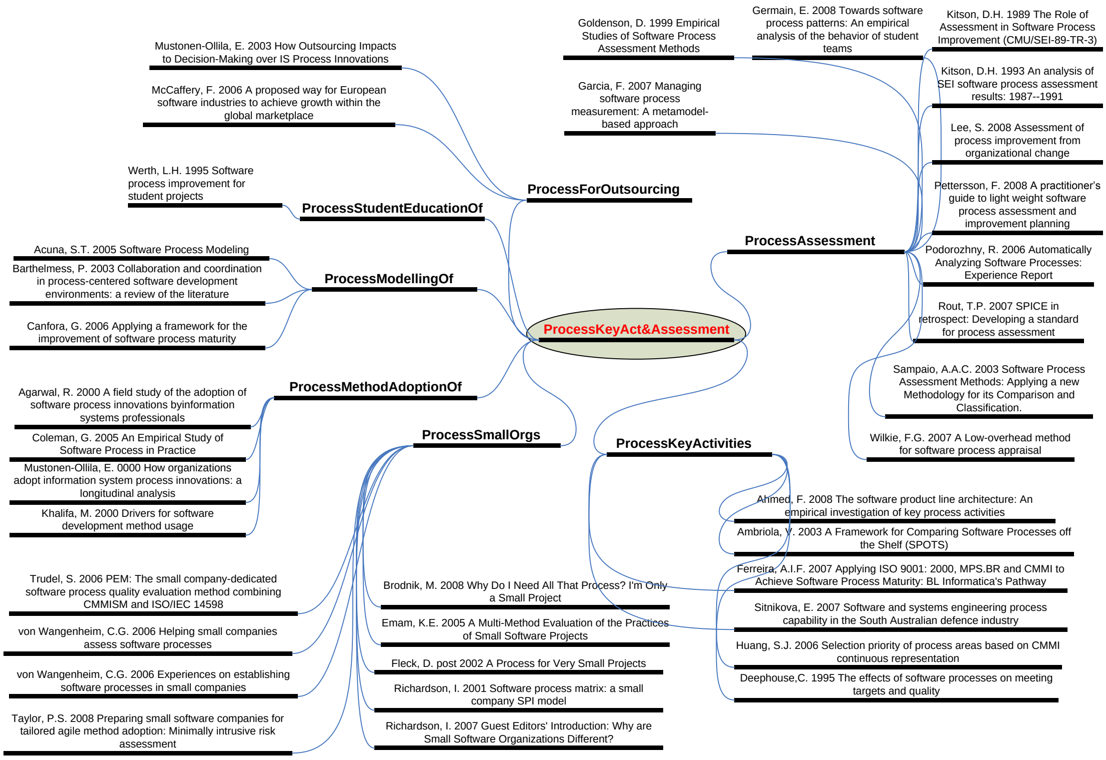
Zahran, S. 1998. *Software process improvement: practical guidelines for business success*. Harlow, Essex, UK: Pearson Education Limited.

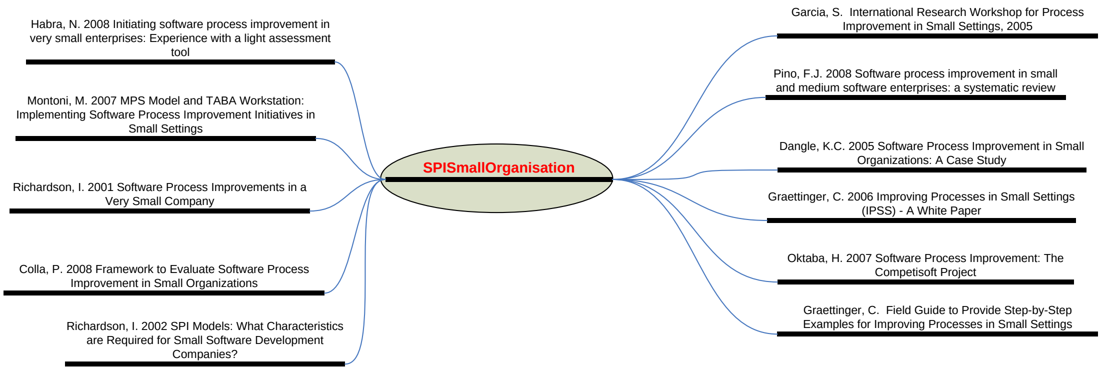
List of Appendixes

Appendix A – Supplementary Literature Review Mind Maps






ProcessKeyAct&Assessment
ProcessForOutsourcing
Mustonen-Ollila, E. 2003 How Outsourcing Impacts to Decision-Making over IS Process Innovations
McCaffery, F. 2006 A proposed way for European software industries to achieve growth within the global marketplace
ProcessStudentEducationOf
Werth, L.H. 1995 Software process improvement for student projects
ProcessModellingOf
Acuna, S.T. 2005 Software Process Modeling
Barthelmess, P. 2003 Collaboration and coordination in process-centered software development environments: a review of the literature
Canfora, G. 2006 Applying a framework for the improvement of software process maturity
ProcessMethodAdoptionOf
Agarwal, R. 2000 A field study of the adoption of software process innovations byinformation systems professionals
Coleman, G. 2005 An Empirical Study of Software Process in Practice
Mustonen-Ollila, E. 0000 How organizations adopt information system process innovations: a longitudinal analysis
Khalifa, M. 2000 Drivers for software development method usage
ProcessSmallOrgs
Trudel, S. 2006 PEM: The small company-dedicated software process quality evaluation method combining CMMISM and ISO/IEC 14598
von Wangenheim, C.G. 2006 Helping small companies assess software processes
von Wangenheim, C.G. 2006 Experiences on establishing software processes in small companies
Taylor, P.S. 2008 Preparing small software companies for tailored agile method adoption: Minimally intrusive risk assessment
Brodnik, M. 2008 Why Do I Need All That Process? I'm Only a Small Project
Emam, K.E. 2005 A Multi-Method Evaluation of the Practices of Small Software Projects
Fleck, D. post 2002 A Process for Very Small Projects
Richardson, I. 2001 Software process matrix: a small company SPI model
Richardson, I. 2007 Guest Editors' Introduction: Why are Small Software Organizations Different?
ProcessAssessment
Goldenson, D. 1999 Empirical Studies of Software Process Assessment Methods
Germain, E. 2008 Towards software process patterns: An empirical analysis of the behavior of student teams
Garcia, F. 2007 Managing software process measurement: A metamodel-based approach
Kitson, D.H. 1989 The Role of Assessment in Software Process Improvement (CMU/SEI-89-TR-3)
Kitson, D.H. 1993 An analysis of SEI software process assessment results: 1987--1991
Lee, S. 2008 Assessment of process improvement from organizational change
Pettersson, F. 2008 A practitioner's guide to light weight software process assessment and improvement planning
Podorozhny, R. 2006 Automatically Analyzing Software Processes: Experience Report
Rout, T.P. 2007 SPICE in retrospect: Developing a standard for process assessment
Sampaio, A.A.C. 2003 Software Process Assessment Methods: Applying a new Methodology for its Comparison and Classification.
Wilkie, F.G. 2007 A Low-overhead method for software process appraisal
ProcessKeyActivities
Ahmed, F. 2008 The software product line architecture: An empirical investigation of key process activities
Ambriola, V. 2003 A Framework for Comparing Software Processes off the Shelf (SPOTS)
Ferreira, A.I.F. 2007 Applying ISO 9001: 2000, MPS.BR and CMMI to Achieve Software Process Maturity: BL Informatica's Pathway
Sitnikova, E. 2007 Software and systems engineering process capability in the South Australian defence industry
Huang, S.J. 2006 Selection priority of process areas based on CMMI continuous representation
Deephouse,C. 1995 The effects of software processes on meeting targets and quality


SPISmallOrganisation
Habra, N. 2008 Initiating software process improvement in very small enterprises: Experience with a light assessment tool
Montoni, M. 2007 MPS Model and TABA Workstation: Implementing Software Process Improvement Initiatives in Small Settings
Richardson, I. 2001 Software Process Improvements in a Very Small Company
Colla, P. 2008 Framework to Evaluate Software Process Improvement in Small Organizations
Richardson, I. 2002 SPI Models: What Characteristics are Required for Small Software Development Companies?
Garcia, S. International Research Workshop for Process Improvement in Small Settings, 2005
Pino, F.J. 2008 Software process improvement in small and medium software enterprises: a systematic review
Dangle, K.C. 2005 Software Process Improvement in Small Organizations: A Case Study
Graettinger, C. 2006 Improving Processes in Small Settings (IPSS) - A White Paper
Oktaba, H. 2007 Software Process Improvement: The Competisoft Project
Graettinger, C. Field Guide to Provide Step-by-Step Examples for Improving Processes in Small Settings


ProjectMgmt& Success
ProjectManagement
ProjectSuccessFailure
2001 Extreme Chaos 2001
Anderson, D.K. 2003 Project Management Strategy—project management represented as a process based set of management domains and the consequences for project management strategy
Beynon-Davies, P. 1999 Human error and information systems failure: the case of the London ambulance service computer-aided despatch system project
Birks, D.F. 2003 Factors influencing GIS project implementation failure in the UK retailing industry
Chaos, E. 2001 The Standish Group International
Donaldson, J. 2001 Workshop on systems failures
Drew Procaccino, J. 2002 Case study: factors for early prediction of software development success
Dvir, D. 2003 An empirical analysis of the relationship between project planning and project success
Gaitros, D. 2004 Common Errors in Large Software Development Projects
Jones, C. 2004 Software Project Management Practices: Success Versus Failure
Larman, C. 2002 How to Fail with the Rational Unified Process: Seven Steps to Pain and Suffering
Larsen, M.A. 1999 When success turns into failure: A package-driven business process re-engineering project in the financial services industry
Lee, S. 2007 Factors affecting the implementation success of Internet-based information systems
Linberg, K.R. 1999 Software developer perceptions about software project failure: A case study
Peterson, D.K. 2002 The perceptions of information systems designers from the United States, Japan, and Korea on success and failure factors
Procaccino, J.D. 2005 What do software practitioners really think about project success: an exploratory study
The Standish Group 1999 The Chaos Report
The Standish Group 1995 The Chaos Report
The Standish Group International, Inc. 1995 Chaos
Verner, J.M. 2005 In-house software development: What project management practices lead to success?
Verner, J.M. 2002 Case study: Factors for early prediction of software development success
Yeo, K.T. 2002 Critical failure factors in information system projects
Bennatan, E.M. 2004 Catastrophe Disentanglement: Getting Software Projects Back on Track
Deephouse, C. 1995 Software processes and project performance
Delaney, C. 1985 The Development of Technical Managers
Desouza, K.C. 2006 Project management offices: A case of knowledge-based archetypes
Garcia, S. 2005 How standards enable adoption of project management practice
McDonald, J. 2002 Software project management audits—update and experience report
McGovern, F. 2002 Managing Software Projects with Business-Based Requirements
Moreno Garcia, M.N. 2004 Building knowledge discovery-driven models for decision support in project management
Murphy, M.G. 1999 Teaching software project management: a response–interaction approach
Napier, N.P. 2008 IT project managers' construction of successful project management practice: a repertory grid investigation
Nidiffer, K.E. 2005 Evolving distributed project management
Peffers, K. 2003 Extending Critical Success Factors Methodology to Facilitate Broadly Participative Information Systems Planning
Pinto, J.K. 2000 Understanding the role of politics in successful project management
Purvis, R.L. 2003 The Impact of Project Management Heuristics to IS Projects
RAE, BCS 2004 The Royal Academy of Engineering and the British Computer Society
Scott, G.M. 2000 Top Priority Management Concerns about New Product Development
Scott, L. 2003 Experience Repositories and the Postmortem
Simmons, L.F. 2002 Project management-critical path method (CPM) and PERT simulated with ProcessModel
Wang, E.T.G. 2008 The effects of change control and management review on software flexibility and project performance
White, D. 2002 Current practice in project management—an empirical study

Appendix B – SPI Activity Survey Instrument

Software Process Improvement (SPI) Activity Survey

Research Team: **Paul Clarke (pclarke@computing.dcu.ie)**
Dr. Rory O'Connor (roconnor@computing.dcu.ie)

Background & Instructions:

This interview session comprises part of a research programme examining small- to medium-sized software development organisations. There are multiple organisations participating in the study – all with strictest confidentiality completely assured.

This survey instrument has been designed as a vehicle for determining the amount of SPI activity in a software development organisation over a period of time. For the purpose of this study, the period of time under investigation is the 12 months preceding the date of survey discharge. Each question in this survey is structured so as to identify any modifications to a specific area of the software development process, with the reference list of process areas being derived from ISO/IEC 12207. The degree of process modification is rated as follows:

0 = No modification
1 = Minor modification
2 = Moderate modification
3 = Significant modification

ISO/IEC 12207 comprises of seven process groups which are further decomposed into forty three smaller process units. These process groupings and units guide the structure of this survey instrument. Where an individual does not have the knowledge to answer any specific question in this survey instrument, the response should be noted as "not known" and every effort should be made to identify an alternative interviewee (from the same organisation) that can provide an informed response.

This research is supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero, the Irish Software Engineering Research Centre (www.lero.ie).

Determine Extent of SPI Activity

The following questions are designed to determine the extent of SPI activity in an organisation over the past year.

1. Software Implementation Processes					
1.1 Regarding **Software Requirements Analysis**, has there been any modification to the approach to:		None	Minor	Moderate	Significant
1.1.1 Specifying requirements, including functionality, consistency, performance and internal interfaces					
1.1.2 Reviewing and testing requirements, confirming the feasibility of design, and examining the impact of requirements on qualification					
1.1.3 Identifying needs such as documentation, installation, acceptance, operation, and future maintenance					
1.1.4 Specifying criteria for safety (e.g. environmental influences, injury), for security (e.g. for sensitive information), and for areas that are sensitive to human errors					
1.1.5 Defining data and databases					
1.1.6 Tracing the origin of and modifications to system requirements, or the general traceability of changes throughout the development process.					
1.2 Regarding **Software Architectural Design**, has there been any modification to the approach to:					
1.2.1 Transforming the requirements into a top level architecture that identifies the hardware and software components					
1.2.2 Developing and documenting the design for external interfaces and databases					
1.2.3 Defining and documenting preliminary test (including integration test) requirements and schedules					
1.2.4 Conducting reviews of the architectural design					
1.3 Regarding **Software Detailed Design**, has there been any modification to the approach to:					

1.3.1 Developing detailed designs for software components, and refining these into low level units that can be coded, compiled and tested					
1.3.2 Developing and documenting detailed designs for internal/external interfaces and databases that permit coding without the need for further information					
1.3.3 Developing, updating and documenting test requirements and schedules, including those that will stress the limits of the system					
1.3.4 Conducting reviews of the detailed design					
1.4 Regarding **Software Construction**, has there been any modification to the approach to:					
1.4.1 Developing and documenting software units and databases, including programming languages and build procedures/tools					
1.4.2 Developing, documenting, executing and maintaining test procedures, and recording test data, for software units and databases					
1.5 Regarding **Software Integration**, has there been any modification to the approach to:					
1.5.1 Developing and executing integration plans with other hardware and software systems, with consideration for responsibilities and schedules					
1.5.2 Developing and documenting a set of test cases and procedures for qualification testing (the process of demonstrating whether an entity is capable of fulfilling specified requirements), and ensuring the software items are ready for qualification testing					
1.5.3 Conducting reviews of software integration efforts					
1.6 Regarding **Software Qualification Testing**, (the process of demonstrating whether an entity is capable of fulfilling specified requirements) has there been any modification to the approach to:					
1.6.1 Conducting qualification testing and documenting the results					

1.7 Regarding **all software development stages**, was there any modification to the process of checking the:					
1.7.1 Feasibility of integration and testing					
1.7.2 Feasibility of operations and maintenance					
1.7.3 Appropriateness of methods and standards					
1.7.4 Internal and external feasibility and consistency of requirements and designs					
1.7.5 Test coverage					
1.7.6 Conformance of testing to expected results					

2. Technical Processes					
2.1 Regarding Requirements Definition, has there been any modification to the approach to:		None	Minor	Moderate	Significant
2.1.1 Identifying stakeholders and their requirements, resolving requirements problems discovered, and establishing agreement with stakeholders that requirements are expressed correctly					
2.2 Regarding **Software Installation**, has there been any modification to the approach to:					
2.2.1 Planning for installation in the target environment and execution of the installation plan					
2.3 Regarding **Software Acceptance Support**, has there been any modification to the approach to:					
2.3.1 Completing and delivering the software product, supporting the acceptance testing and reviewing activities, and providing training and support as specified in contract					
2.4 Regarding **Software Operation and Maintenance**, has there been any modification to the approach to:					
2.4.1 Testing the system in its operational environment, ensuring correct operation as per user documentation and thereafter, activating					

the system usage					
2.4.2 Planning for operation and maintenance, including defining and/or documenting the procedure for releasing software, for problem report and modification request submission and resolution (incl. configuration management)					
2.4.3 Migrating systems to new environments, including the collection of associated requirements, system verification (including any required updates), parallel operation of environments, and the maintenance of data associated with old environments					
2.5 Regarding **Software Disposal**, has there been any modification to the approach to:					
2.5.1 Defining and executing a software disposal strategy, which may include the notification of associated activities, and the control of access to data associated with retired software products in accordance with contract and data protection/audit requirements					

3. Software Support Processes

3.1 Regarding Software Configuration Management, has there been any modification to the approach to:		None	Minor	Moderate	Significant
3.1.1 Developing and documenting a configuration management plan that covers: procedures, activities and schedules; identifying responsible organisation(s), and supporting an audit trail					
3.1.2 Identifying software items for configuration management, using baselines, and deciding which versions are controlled for which projects					
3.1.3 Performing change request (CR) activities which may include: the identification and recording of CRs; the analysis and evaluation of changes; the approval/disapproval of CRs; the implementation, verification and release of modified software					
3.2 Regarding **Problem Resolution**, has there been any modification to					

the approach to:					
3.2.1 Handling problems detected in the software products and processes, which may include the use of problem reports, the tracking and reporting of issues and resolutions, and the analysis of trends					
3.3 Regarding **Software Documentation Management**, has there been any modification to the approach to:					
3.3.1 Producing, reviewing, maintaining and controlling documents throughout the various stages of the lifecycle					
3.4 Regarding **Software Quality Assurance**, (all the planned and systematic activities implemented within the quality system) has there been any modification to the approach to:					
3.4.1 Planning, defining, resourcing, standardising and executing QA activities so as to assure that products and processes comply with established processes and as required by contracts; possibly using measurements and applied to subcontractors as applicable					
3.5 Regarding **System Validation** (validating that the right product being built), has there been any modification to the approach to:					
3.5.1Using of methods, techniques and tools to conduct validation tests, including stressful and failure scenarios, and fit for purpose testing, to validate that the system satisfies its intended use in the target environment					
3.6 Regarding **Software Reviewing**, has there been any modification to the approach to:					
3.6.1 Reviewing process implementation, project management or technical reviews of products					
3.7 Regarding **Software Audits**, (an internal/external independent assessment of software products and processes) has there been any modification to the approach to:					
3.7.1 Conducting independent audits of projects or software deliverables, including examining that coded products reflect the design, that the product fulfils the requirements and that costs and schedules adhere to plans.					

4. Software Reuse Processes					
4.1 Regarding the Reuse of Software Assets, has there been any modification to the approach to:		None	Minor	Moderate	Significant
4.1.1 Managing and controlling reusable domain assets, including the handling of problem reports and change requests for assets such that problems/modifications are reviewed for conformance across the broader spectrum of application of the assets					
4.2 Regarding **Reuse Programme Management**, has there been any modification to the approach to:					
4.2.1 Programme-level planning for reuse, including the identification of domains for reuse or refining/re-scoping existing domains in light of future software product plans					

5. Organisation Project-Enabling Processes					
5.1 Regarding Life Cycle Management, has there been any modification to the approach to:		None	Minor	Moderate	Significant
5.1.1 Selecting, tailoring, monitoring, controlling and assessing life cycle processes / standards / methods / models.					
5.1.2 Collecting and maintaining quality cost data					
5.2 Regarding **Project Portfolio Management**, has there been any modification to the approach to:					
5.2.1 Initialising and authorising projects, which may include the identification of accountabilities, outcomes and multi-project interfaces, the allocation of resources and the reporting/review requirements					
5.3 Regarding **Infrastructure Management**, has there been any modification to the approach to:					
5.3.1 Planning, documenting, providing,					

maintaining and configuring an infrastructure which may include: hardware, software and facilities for development, operation and maintenance					
5.4 Regarding **Human Resource Management**, has there been any modification to the approach to:					
5.4.1 Reviewing organisation and project requirements for the provision or development of resources and skills, including the types and levels of training and knowledge required and the recruitment of new staff					
5.4.2 Setting up and empowering teams that are equipped with the right mix of personnel					
5.4.3 Planning for knowledge management, which may include the establishment of a network of experts and the configuration management of knowledge assets					
5.5 Regarding **Quality Management**, has there been any modification to the approach to:					
5.5.1 Managing quality using policies and procedures which identify responsibilities and authority, and that are aligned with business goals and objectives					
5.5.2 Reviewing quality policies and procedures, assessing customer satisfaction, and taking corrective action when quality management goals are not achieved					

6. Project Processes

6.1 Regarding Project Planning, has there been any modification to the approach to:		None	Minor	Moderate	Significant
6.1.1 Project initiation, which may include the setting of project objectives, the establishment of feasibility (e.g. resources), establishing the achievability of timescales					
6.1.2 Preparing project plans, which may include: schedules, effort estimation, resources, task allocation, responsibility assignment, risk quantification, quality					

assurance measures, environment provision and life cycle model definition					
6.2 Regarding **Project Assessment & Control**, has there been any modification to the approach to:					
6.2.1 Reviewing and reporting on the execution of projects and products, maintaining control of projects and resolving instances of lack of progress					
6.2.2 Closing projects when all tasks are complete as per specified criteria (or when a project is cancelled), archiving results and records in appropriate environment(s)					
6.3 Regarding **Risk Management**, has there been any modification to the approach to:					
6.3.1 Risk management planning and control, including risk management process definition, and the continual monitoring of existing and new risks, and implementing risk treatments					
6.4 Regarding **Configuration Management**, has there been any modification to the approach to:					
6.4.1 Defining a general configuration management strategy, identifying items that are subject to configuration control (beyond software configuration management), maintaining these controlled items, ensuring appropriate levels of integrity and security					
6.5 Regarding **Information Management**, has there been any modification to the approach to:					
6.5.1 Obtaining, maintaining, archiving and disposing of information items, for example confidential client data, according to knowledge retention, audit, integrity, policy, security and privacy requirements					
6.6 Regarding **Measurement Management**, has there been any modification to the approach to:					
6.6.1 Identifying and evaluating information needs, for example project progress and product quality measurements,- including the procedure(s) for data collection,					

analysis, verification and reporting					

7. Agreement Processes					
7.1 Regarding Software Acquisitions, has there been any modification to the approach to:		None	Minor	Moderat	Signific ant
7.1.1 Preparing for the acquisition of software components (requirements definition and RFPs), evaluating candidate suppliers for the acquisition of software components, and thereafter, selecting and monitoring suppliers of software components.					
7.2 Regarding **Software Supply**, has there been any modification to the approach to:					
7.2.1 Identifying opportunities, tendering for business including bidding and negotiating contracts, and managing changes through change control					

Interview Round-up

1. Of all the process modifications that have been discussed, can you list your top three most important process changes?

1. 2. 3.

2. In your view, has this survey covered all of the process areas that are relevant to your organisation or can you identify processes that have not been covered in this survey?

3 Do you have any other closing comments that you would like to make?

Appendix C – Situational Change Survey Instrument

Situational Change Survey

Research Team:	**Paul Clarke (pclarke@computing.dcu.ie)** **Dr. Rory O'Connor (roconnor@computing.dcu.ie)**

Background & Instructions:

This interview session comprises part of a research programme examining small- to medium-sized software development organisations. There are multiple organisations participating in the study – all with strictest confidentiality completely assured.

This survey instrument has been designed as a vehicle for determining the amount of situational change in a software development organisation over a period of time. In particular, this survey instrument is utilised to determine the extent changes to aspects of situational context that are known to affect the software development process. For the purpose of this study, the period of time under investigation is the 12 months preceding the date of survey discharge. Each question in this survey is structured so as to identify any modifications to a specific situational factor that affects the software development process. The degree of situational modification is rated as follows:

0 = No modification
1 = Minor modification
2 = Moderate modification
3 = Significant modification

The reference list of situational factors affecting the software development process has derived from a grounded theory based approach to synthesising the factors of situation as presented in a variety of related domains, including: software cost estimation, software development models and standards, and software risk management.

Where an individual does not have the knowledge to answer any specific question in this survey instrument, the response should be noted as "not known" and every effort should be made to identify an alternative interviewee (from the same organisation) that can provide an informed response.

This research is supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero, the Irish Software Engineering Research Centre (www.lero.ie).

Determine Extent of Situational Change

The following questions are designed to determine the extent of situational changes (which impact on the software development process) in an organisation over the past year.

1. Nature of Application(s)/Product(s) under Development					
1.1 Regarding the application(s)/product(s) under development, has there been any modification to:		None	Minor	Moderate	Significant
1.1.1 The required performance of the application(s)/product(s) – including reliability and real time performance of both hardware and software components?					
1.1.2 The application(s)/product(s) type, for example application domain, criticality, and backup and recovery demands?					
1.1.3 The complexity of the application(s)/product(s) including inherent application/product complexity and hardware architecture complexity?					
1.1.4 The application(s)/product(s) size, including required storage and relative project size?					
1.1.5 The number of links from the application(s)/product(s) to external systems?					
1.1.6 The volatility of the application(s)/product(s), for example, extent of recent application(s) changes?					
1.1.7 Degree of risk associated with the application(s) – for example, changes to the way people work as a result of changes to or new application(s)/product(s)?					
1.1.8 The number of different versions of application(s)/product(s) requiring support?					

2. Personnel					
2.1 Regarding personnel, has there been any modification to:		None	Minor	Moderate	Significant
2.1.1 Relative team size?					
2.1.2 Personnel turnover?					
2.1.3 Personnel productivity, including the ability to carry out tasks quickly?					
2.1.4 Personnel skill levels, including, operational knowledge and expertise/capability to complete allocated tasks?					
2.1.5 Team cohesion, for example, the duration the team has worked together, the distributed nature of the team, and dependence on individual team members?					
2.1.6 The experience of personnel, including application(s) /product(s) experience, platform experience, and development methodology experience?					
2.1.7 Personnel culture, including team culture and resistance to change?					
2.1.8 Personnel commitment to ongoing work/project activities?					
2.1.9 Personnel disharmony, for example interpersonal conflicts?					

3. Characteristics of Requirements					
3.1 Regarding the characteristics of requirements, have there been any modifications to:		None	Minor	Moderate	Significant
3.1.1 The changeability of the requirements, including unclear and changing requirements?					
3.1.2 The feasibility of the requirements, for example, do they strain present capabilities?					
3.1.3 The quality of requirements, including their correctness and definition?					
3.1.4 The rigidity of requirements, for instance, the extent to which requirements compliance is mandated?					

4. Organisation

4.1 Regarding organisation, have there been any modifications to:		None	Minor	Moderate	Significant
4.1.1 Organisational stability, for example, rate of organisational growth or decline, and constant shifting of resources between projects?					
4.1.2 The organisational facilities, including the office space, personal space of staff, and supporting IT infrastructure?					
4.1.3 The overall size of the organisation? For example, changes in the headcount.					
4.1.4 The commitment of the senior management team to projects?					
4.1.5 The organisational maturity, including the steady progress of planned work?					
4.1.6 The organisational structure, for example different divisions of staff and responsibilities?					

5. Operations

5.1 Regarding operations, have there been any modifications to:		None	Minor	Moderate	Significant
5.1.1 The commitment of end-users to the product(s) or application(s)?					
5.1.2 The level of disharmony observed between end-users?					
5.1.3 The volume or profile of end-users?					
5.1.4 The turn-over of end-users?					
5.1.5 The expertise of end-users?					
5.1.6 Operation pre-requisites, including applicable standards and laws?					
5.1.7 The required ease of operation or installation of application(s) or product(s)?					

6. Business					
6.1 Regarding the business, have there been any modifications to:		None	Minor	Moderate	Significant
6.1.1 External dependencies, including outside suppliers for hardware, software or other services?					
6.1.2 External dependencies, including multiple suppliers on large projects and reliance on other projects or processing systems?					
6.1.3 The number of external stakeholders?					
6.1.4 Key business drivers, including maximising or minimising turnover, profit and costs; and market conditions (growing/declining)?					
6.1.5 The need to accelerate the time-to-market of new features or products or applications?					
6.1.6 The degree of customer satisfaction?					
6.1.7 The payment terms for projects, including time and materials, fixed price, and phase payments?					
6.1.8 The opportunities offered by projects, including possible loss leaders into new clients or markets?					
6.1.9 The potential loss accruing from project failures, including customer relations, financial health, organisation survival, and loss of human life?					

7. Technology					
7.1 Regarding the technology, have there been any modifications to:		None	Minor	Moderate	Significant
7.1.1 The extent of use of emerging or immature technology?					
7.1.2 The knowledge of technology, including new language, tools, hardware and software?					

8. Management

8.1 Regarding management, have there been any modifications to:		None	Minor	Moderate	Significant
8.1.1 The degree of management accomplishment, including project management experience and operational knowledge?					
8.1.2 The composition and size of the management team?					
8.1.3 Management expertise, including project management, planning, estimation people skills?					
8.1.4 Management expertise, including the ability to manage clients and suppliers?					

Interview Round-up

1. Of all the situational modifications that have been discussed (Nature of application(s)/product(s) under development, Personnel, Characteristics of requirements, Organisation, Operation, Business, Technology, and Management), can you list your top three most important changes?

1. 2. 3.

2. In your view, has this survey covered all of the situational aspects that are relevant to your organisation or can you identify other situational aspects **that potentially affect the software development process** that have not been covered in this survey?

3 Do you have any other closing comments that you would like to make?

Appendix D – Business Success Survey Instrument

Investigating the influence of software process improvement on business success in SMEs

Research Project Lead & Interviewer: Paul Clarke (pclarke@computing.dcu.ie)

Background:

This interview session comprises part of a research programme examining small- to medium-sized software development organisations. There are multiple organisations participating in the study – all with strictest confidentiality completely assured.

The purpose of the study is to examine the extent to which the entire software development process is managed with respect to the emerging needs of the organisation, and to investigate the impact that this process management activity has on business success. In order to carry out the research, two separate channels of inquiry are required in each participating company.

The first channel of inquiry examines the extent of achievement of business objectives – and this requires two interviews of approximately 30 minutes each with a member of the executive management team. The first interview with the executive team will determine the broad, high level business objectives for the following 12 months. The second interview will take place approximately one year after the initial interview and will re-visit the business objectives and establish the extent to which they have been achieved.

The second channel of inquiry examines the software process management activities carried out in the businesses under study. This involves an interview with a senior member of the software development team of approximately 2 hours duration, and is used to determine the extent of software process improvement activities in the firm and to examine the estimated need for software process improvement.

This work is supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero, the Irish Software Engineering Research Centre (www.lero.ie).

General Information

Organisation and Interviewee	
Company Name	
Sector (Product dev for)	
Interviewee	
Current Position	
Interview Date	
Interview Location	
Start Time	
End Time	

Organisation Size – these questions are required so that the status of "SME" can be confirmed.	
Does the number of employees exceed 249? (probably need an exact number – also needed for the engineering function so as to allow for size profiling)	
Can the exact number of employees be revealed for this interview?	
Does the annual turnover exceed €50 million?	
Does the annual balance sheet total exceed €43 million?	

Determine Future Company Objectives

The following questions are designed to determine the business objectives for the organisation for the forthcoming year. In accordance with the Holistic Scorecard framework for performance determination for software development companies, the business goals are categorised under six different classifications: financial, customer, business process, intellectual capital, employee and social.

Where the question is not applicable or the response is not available "N/A" is recorded.

Where an objective remains unchanged from the current year, "S/Q" (Status Quo) is recorded.

1. Financial Objectives	
1.1 Compared with the year just passed, what is the relative target (%age growth or decline) for:	
1.1.1 Revenue	
1.1.2 Profit	
1.1.3 Assets (not including intellectual capital)	
1.2 Other than revenue, profit and assets, list the relative targets for any other financial measures (e.g. return on investment):	
1.2.1	
1.2.2	
1.2.3	

2. Customer Objectives
2.1 For the forthcoming year, what is the target for:

2.1.1 Number of new client acquisitions	
2.1.2 Profile of new client acquisitions (larger, smaller, same)	
2.1.3 Number of existing clients presenting for repeat business	
2.1.4 Market share in core and allied markets (%age growth or decline)	
2.2 For the forthcoming year, are there any objectives in relation to:	
2.2.1 Extending the product offering through the addition of new product features or through the development of new product offerings	
2.3 Compared with the year just passed, what is the relative target for:	
2.3.1 Customer satisfaction (make the customers more or less satisfied, or status quo)	
2.3.2 Brand image	

3. Business Process Objectives	
Regular Business Processes	
3.1 For the forthcoming year, list any objectives that exist in relation to:	
3.1.1 Business process management	
3.1.2 Changing the nature of customer interaction	

3.1.3 Organisational productivity	
3.1.4 Seeking / Retaining a recognised quality standard (e.g. ISO)	
3.1.5 Effectiveness of business documentation	
3.1.6 Effectiveness of business tools and techniques	
3.1.7 Organisational focus on continuous improvement or benchmarking against best practice	
Risk Management Process	
3.2 For the forthcoming year, list any objectives that exist in relation to:	
3.2.1 Risk management planning, and the process for risk identification, analysis, monitoring and control (risks items could include: quality/performance risks, project management risks)	
Knowledge Management Process	
3.3 For the forthcoming year, list any objectives that exist in relation to:	
3.3.1 The diffusion of knowledge throughout the organisation	
3.3.2 Developing or extending internal knowledge repositories (e.g. experience databases or expertise locators)	

4. Intellectual Capital Objectives	
Human Capital – cerebral aspects of the individual people in the organisation	
4.1 For the forthcoming year, list any employee objectives that exist in relation to:	
4.1.1 Employee competence & expertise	
4.1.2 Skill sets	
4.1.3 Employee qualifications	
4.1.5 Employee Experience	
4.1.6 Intellectual Agility	
4.1.7 Attitude To Work	
Organisational Capital – cerebral aspects of the organisation as a whole	
4.2 For the forthcoming year, list any objectives that exist in relation to:	
4.2.1 Patents, copyrights, branding and trade secrets	
4.2.2 Registered designs and processes	
4.2.3 Research publications	

5. Employee Objectives	
Recruitment, Selection and Retention	
5.1 For the forthcoming year, list any objectives that exist in relation to:	
5.1.1 Organisational Headcount	
5.1.2 The effectiveness of recruitment and selection procedures	
5.1.3 Retention strategies (especially for precious resources)	
5.1.4 Structured and rewarding career paths	
Training and Education	
5.2 For the forthcoming year, list any objectives that exist in relation to:	
5.2.1 Training in technical, hard and soft skills, and new technologies	
5.2.2 Training effectiveness	
5.2.3 The alignment of training programmes with long term organisational goals	
Company Culture	
5.3 For the forthcoming year, list any objectives that exist in relation to:	
5.3.1 Co-operation and co-ordination among people at different levels, functions and departments	
5.3.2 Communication effectiveness	
5.3.3 Change management (i.e. how to chaperone employees through change events)	

Employee Satisfaction	
5.4 For the forthcoming year, list any objectives that exist in relation to:	
5.4.1 Employee empowerment	
5.4.2 Career growth	
5.4.3 Pay and perks	
5.4.4 Rewards / Recognitions	
5.4.5 Redressing grievances	

6. Social Objectives – highly abstract, related to the quality and value of relationships enjoyed with larger society through the exercise of corporate citizenship	
Political Image	
6.1 For the forthcoming year, list any objectives that exist in relation to:	
6.1.1 Compatibility with regulatory bodies and local government	
6.1.2 Aid and subsidies, infrastructural support, tax exemptions, special favours, etc., from government	
6.1.3 Active membership in national and international software associations, institutions, societies and communities	
Social Image	
6.2 For the forthcoming year, list any objectives that exist in relation to:	
6.2.1 Corporate citizenship (self-regulation with respect to legal and ethical standards)	
6.2.2 Contribution to the society (e.g. charity work and employment opportunities for the less privileged)	
6.2.3 Goodwill, general image among the common public	

Interview Close Out

The following questions are designed to determine the order of priority attached to each of the six perspectives of the Holistic Scorecard.

Priority of Holistic Scorecard Perspectives		
Perspective	**In order of Priority**	**Comment**
Financial		
Customer		
Business Process		
Intellectual Capital		
Employee		
Social		

<u>Q. Can the interviewee identify the top three priorities (in order) for the forthcoming year?</u>

1.
2.
3.

<u>Q. Can the interviewee identify any business objectives that have not covered by this Interview?</u>

<u>Closing comments from interviewee:</u>

<u>Closing comments from interviewer:</u>

Follow up in 10 months, speak with a member of the engineering team to determine how successful the organisation has been with respect to software process management. Who might this person be? Perhaps they can be given a head's up at this stage?

And of course – many thanks for the time taken to participate in the study.

Appendix E – Analysis of Related Situational Factors Publications

Analysis of framework when compared with Dede and I. Lioufko. 2010. *Situational factors affecting software development process selection.* PhD. Master of Science, University of Gothenburg:

Factors identified	Covered already?	Category	Factor(s)	Sub-Factor(s)
Team size	Y	Personnel	Team size	(Relative) team size
Working experience of team members	Y	Personnel	Experience	(General) experience, Application experience, Platform experience, Analyst experience, Programmer experience, Tester experience
Education and awareness of team members on frameworks, tools and strategies	Y	Personnel, Technology	Skill, Knowledge	Training development team members, Experience with development methodology, Platform experience, Language experience, Tools experience, Experience with (general) technology
Domain knowledge transfer	Y	Application	Type	Application domain
Team distribution	Y	Personnel	Cohesion	Distributed team
Multicultural team	Y	Personnel	Cohesion	(General) cohesion, Team geographically distant
Project duration	Y	Application	Application Size	Relative project size and duration
Project size	Y	Application	Team Size, Application Size	(Relative) team size, Relative project size and duration
Acceptance of requirements changes	Y	Requirements	Changeability	Scope creep, Continually changing system requirements, Unclear system requirements
Quality level	Y	Application	Quality	Required product quality
Monitoring process procedure	Y	Management	Expertise	Progress control capability

Personal effect on working practices	Y	Organisation, Management	Maturity, Management Commitment	Maturity of the organisation, Unstable organisational environment, Effectiveness of project management methodology, Effectiveness of work flow and coordination
Certification	Y	Operation, Management, Application	Prerequisites, Expertise, Application	Applicable standards, Applicable laws, Organisational policies, Effectiveness of project management methodology, Project planning capability, Project management systems, Application domain, Application criticality, Application type
Effectiveness of resource usage	Y	Management	Expertise	Project planning capability, Achievability of schedules and budgets, Estimation capability with respect to the personnel needs of projects, Effectiveness of work flow and coordination.
Business domain	Y	Operation, Application, Business	Prerequisites, Application type, Business drivers, Opportunities, Time to market, Magnitude of potential loss	Applicable standards, Applicable laws, Organisational policies, Application domain, Project drivers, Marketing considerations, Time to market, Loss of human life
Contract type	Y	Business	Payment arrangements, External dependencies	Time and materials, Fixed price, Multiple implementers, Number of involved parties
Customer /Stakeholder involvement	Y	Business	External dependencies	Number of (external) stakeholders, Stakeholders' background, Access to stakeholders

Analysis of framework when compared with Bekkers, W., van de Weerd, I., Brinkkemper, S. and Mahieu, A. 2008. The Influence of Situational Factors in Software Product Management: An Empirical Study. *IN: Proceedings of the Second International Workshop on Software Product Management (IWSPM '08),* Los Alamitos, CA, USA: IEEE Computer Society:

Factors identified	**Covered already?**	**Category**	**Factor(s)**	**Sub-Factor(s)**
Development philosophy	Y	Organisation, Personnel, Requirements	Maturity, Experience, Changeability	Use of modern programming practices, Experience with development methodology, Continually changing system requirements, Unclear system requirements
Size of business unit team	Y	Organisation	Organisation Size	Size of the organisation
Size of development team	Y	Personnel, Application	Team Size, Application Size	(Relative) team size, Relative project size and duration
Customer loyalty	Y	Operation, Business	End-Users, Customer Satisfaction	End user commitment, Customer satisfaction
Customer satisfaction	Y	Business	Customer Satisfaction	Customer Satisfaction
Customer variability	Y	Operation	End-Users	User turnover, End-user familiarity with application type
Number of customers	Y	Application, Operation	Degree of risk, End-Users, Deployment Profile	Number of people the project affects, Number of departments affected, Number of users in organisation, Number of deployed versions of applications, Number of deployed applications
Number of end-users	Y	Application	Degree of risk, End-Users	Number of people and departments the project affects, Number of departments affected, Number of users in organisation
Type of customers	Y	Operation	End-Users	End-user familiarity with application type, End-user understanding of system capabilities and

				limitations
Hosting demands	Y	Application, Organisation	Type, Complexity, Facilities	Application type, Architecture type, Product Complexity, Hardware Architecture, Facilities to house the project
Localisation demand	Y	Application, Operation	Type, End-Users	Application type, Architecture type, Configuration demands, Number of users outside the organisation
Market growth	Y	Organisation	Stability, Maturity	Rate of organisational change (growth or decline), Maturity of the organisation
Market size	Y	Organisation	Stability, Maturity	Rate of organisational change (growth or decline), Maturity of the organisation
Release frequency	Y	Application, Management	Deployment profile, Expertise	Number of deployed versions of applications, Number of deployed applications, Achievability of schedules and budgets, Effectiveness of work flow and coordination
Sector	Y	Application, Business	Type, Business Drivers	Application type, Application domain, Application criticality, Marketing activities
Standard dominance	Y	Operation, Requirements	Prerequisites, Rigidity	Applicable standards, Applicable laws, Rigidity of compliance to requirements
Variability of feature requests	Y	Requirements	Changeability	Scope creep, Continually changing system requirements, Unclear system requirements
Application age	Y	Application	Deployment Phase	Development, Maintenance, Other phases (e.g. prototyping, long term maintenance, end of life)
Defects per year	Y	Application	Quality	Required product quality

Development platform maturity	Y	Technology, Application	Knowledge, Emergent, Development Phase, Reuse	Experience with (general) technology, Emerging technology, Development, Maintenance, Other phases (e.g. prototyping, long term maintenance, end of life), Required reuse
New requirements rate	Y	Requirements	Changeability	Scope creep, Continually changing system requirements, Unclear system requirements
Number of products	Y	Application	Deployment Profile	Number of deployed applications
Product lifetime	Y	Application	Deployment Phase	Development, Maintenance, Other phases (e.g. prototyping, long term maintenance, end of life)
Product size	Y	Application	Application Size, Connectivity	Hardware, Software, Required storage, Number of links to existing systems
Product tolerance	Y	Application	Type, Predictability, Complexity	Backup and recovery demands, Extent of recent changes, Platform volatility, Required reliability
Software platform	Y	Application	Reuse, Complexity	Extend of use of externally-sourced components, Product complexity, Required reliability
Company policy	Y	Operation	Prerequisites	Organisational policies, Applicable standards
Customer involvement	Y	Business, Requirements	External Dependencies, Standard	Number of involved parties, Engagement of end-users in requirements capture, User understanding of requirements
Legislation	Y	Operation	Prerequisites	Applicable laws
Partner involvement	Y	Business, Requirements	External Dependencies, Standard	Number of involved parties, Dependency on outside suppliers, Engagement of end-users in requirements capture, User understanding of requirements

Analysis of framework when compared with Petersen, K. and Wohlin, C. 2009. Context in industrial software engineering research. *IN: Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement,* Washington, DC, USA: IEEE Computer Society:

Factors identified	**Covered already?**	**Category**	**Factor(s)**	**Sub-Factor(s)**
Maturity	Y	Technology, Application	Knowledge, Emergent, Development Phase, Reuse	Experience with (general) technology, Emerging technology, Development phase, Maintenance phase, Other phase(e.g. prototyping, long term maintenance, end of life), Required reuse
Quality	Y	Application	Quality	Required product quality
Size	Y	Application	Application Size, Connectivity	Hardware, Software, Required storage, Number of links to existing systems
System type	Y	Application	Type	Application type, Application domain
Customisation	Y	Application	Type	Application type, Architecture type, Configuration demands
Programming language	Y	Technology	Knowledge	Language experience, Experience with (general) technology
Activities	Y	Personnel	Experience, Organisation	Experience with development methodology, Use of modern programming practices
Work-flow	Y	Management	Expertise	Effectiveness of project management methodology, Effectiveness of work flow and coordination
Artifacts	Y	Management	Expertise	Project management systems, Experience with management tools
CASE tools	Y	Technology	Knowledge	Tools experience
Practices and techniques	Y	Organisation	Maturity	Use of modern programming techniques
Roles	Y	Management	Expertise	Effective understanding

				of responsibilities
Experience	Y	Personnel, Management	Experience, Skill, Expertise, Accomplishment, Continuity	(General) experience, Team experience, Application experience, Analyst experience, Programmer experience, Tester experience, Experience of development methodology, Platform experience, Training development team members, Experience with project management tools, Project management experience, Operational knowledge of leader, Changes in organisational management
Model of overall organisation	Y	Organisation	Maturity, Structure	Maturity of the organisation, Inappropriate structure
Organisational unit	Y	Personnel	Team Size, Cohesion, Management Commitment, Organisational Size	(Relative) team size, (General) cohesion, Team not having worked together in the past, Senior management commitment to project, Size of the organisation
Certification	Y	Operation, Management, Application	Prerequisites, Expertise, Application	Applicable standards, Applicable laws, Organisational policies, Effectiveness of project management methodology, Project planning capability, Project management systems, Application domain, Application criticality, Application type
Distribution	Y	Personnel	Cohesion, Culture	Distributed team, Team geographically distributed, Team culture
Number of customers	Y	Application, Operation, Business	Deployment Profile, End-Users, Business Drivers	Number of deployed versions of applications, Number of deployed applications, Number of users outside the organisation, Number of

				users in organisation, Project drivers
Market segments	Y	Application, Business	Type, Business Drivers, Time to Market, Customer Satisfaction	Application type, Application domain, Project drivers, Marketing activities, Time to market, Customer relations
Strategy	Y	Application, Business	Quality, Type, Business Drivers	Required product quality, Application type, Project drivers
Constraints	Y	Business, Operation, Management, Application	Time to Market, Prerequisites, Type	Time to Market, Applicable standards, Applicable laws, Organisational policies, Application domain, Application criticality, Application type