A REQUIREMENTS-BASED SOFTWARE PROCESS MATURITY MODEL

Sarah Beecham

A thesis submitted in partial fulfilment of the requirements of the University of Hertfordshire for the degree of Doctor of Philosophy

The programme of research was carried out in the Department of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire

December 2003

CONTENTS

PART TWO: REQUIREMENTS ENGINEERING PROBLEMS

PART THREE: REQUIREMENTS ENGINEERING SOLUTIONS

PART FOUR: CONCLUSIONS

List of Figures

Page No.

List of Tables

Page No.

Acknowledgements

I dedicate this thesis to Peter, Emma and Nick – my very special family – who I love very much. Without their love, faith and support I could not have written one word of this thesis.

I am very grateful to the University of Hertfordshire for funding my PhD program and for giving me the opportunity to work in such an interesting field for three years. I feel very privileged to have been a post graduate student in the Department of Computer Science where I was introduced to the exciting world of research. Very great thanks go to my principal supervisor, Tracy Hall, for her inspiration, guidance and constant encouragement throughout my study. I cannot imagine a better supervisor. I also thank my supervisory team members, Carol Britton for her clear thinking and guidance, and Michaela Cottee from the Business School for her great help with my methodology and for giving a different perspective to the work. I am also grateful to Austen Rainer for his valuable advice and support. Working in the same office as Austen and Nathan Baddoo when I began my research, gave me such a boost – their total dedication to research, respect and friendship has influenced me greatly.

My experience at UH would not be complete without thanking my lunchtime 'bridge' friends for their support. Thanks also go to Sue Anthony for allowing me to sit in on her statistics lectures. Also, thanks to my friends in the office Rizwan Ahmed, Nilay Oza, Jeni Swan – working from Norwich was made a lot easier knowing I could count on them. Thanks also to all my Hertfordshire friends – (in no particular order!) Rob and Sue, Theresa and Tony, Sarah, Fiona, Dave and Maggie, Nick and Sue for giving me respite. I am very grateful to Marion Catlin for proof reading parts of my work.

I also thank all the experts who gave their time to validating the model and to the companies who collaborated in the software process improvement project.

Last, but definitely not least I extend my gratitude to my mother for her faith in me, to my sister Maria for her positive attitude and love, to my sister Sonia for her great strength, high standards and care, to my brother Ben for his youthful enthusiasm and to my brother Rick for making me laugh.

Abstract

The requirements phase of software development is an on-going problem for the software engineering community. The many disparate recommendations and best practices found in the literature make it difficult for software organisations to recognise which practices apply to their individual needs. The aim of this thesis is to pull together key solutions into a framework that allows practitioners to assess where their requirements process needs strengthening and to provide a means in which improvements can be achieved.

In this thesis I show how I design, develop and validate a model of requirements engineering processes. This requirements capability maturity model (R-CMM) adheres to the characteristics of the Software Engineering Institute's Software Capability Maturity Model (SW-CMM) and is designed to take practitioners from an immature process capability through to an advanced capability.

I contribute to the body of knowledge in both software process improvement and requirements engineering (RE) by providing rigorous detail of how a process maturity framework is developed to support RE practices. The model is generic and should apply to many software development organisations. The R-CMM guides users towards a view of RE that is based on goals and is problem driven. The SW-CMM framework is transformed into a simplified model that relates goals and problems to individual RE practises.

PART ONE

BACKGROUND

Chapter One: Introduction

This thesis presents a validated requirements capability maturity model (R-CMM) that aims to support practitioners in their software process improvement activities. This requirements engineering process model provides a pathway to improved processes by prompting practitioners to examine requirements engineering processes within a five-stage maturity framework. At the current stage of model development, the initial maturity stages are developed in depth in order to gauge where future work is required to complete the model. This thesis describes the rigorous methods used in the development of the model from inception through to validation.

The R-CMM developed in this thesis is based on an empirical study of requirements engineering process problems experienced by software practitioners in the UK today. Although all companies in the study were using the Software Capability Maturity Model (SW-CMM) (Paulk et al. 1995) to guide them in their software process improvement activities, they all showed a lack of control over their requirements engineering process (Beecham et al. 2003d).

This study builds on data collected in the 'Managing Practitioners' Impact on Process and Product' (PPP) project. The PPP project was funded by the Engineering and Physical Sciences Research Council (EPSRC) under grant number GR/L91962. The PPP project investigated human aspects of software process improvement (SPI) implementation in UK companies. It explored a variety of issues from practitioners' understanding of SPI, skills for SPI, the interpersonal politics involved in SPI and motivators and de-motivators for SPI. It is in the research reported in this thesis, however, that the problems practitioners are experiencing with their requirements engineering (RE) process are investigated and a model of the RE process is proposed and validated.

The literature shows the RE process to be notoriously difficult to manage. Practitioners are not short of solutions to their many requirements problems, but need a framework in which they can recognise weaknesses and prioritise their individual needs. This study differentiates itself from other work in the field by placing requirements processes in context with the SW-CMM. Basing the R-CMM on a known software improvement framework offers the user many advantages. The R-CMM taps into the strengths of the SW-CMM to form a specialised best practice model that is familiar, integrates with related software processes, and has a tried and tested methodology. The framework offered by the SW-CMM pulls together disparate work in the field of the RE process and presents solutions in a way that is accessible to both practitioners and researchers. The R-CMM includes an assessment method that guides the user to identify strengths and weaknesses in their current requirements process with a view to prioritizing process implementation against maturity goals.

1.1 An overview of the requirements engineering process

Over the past twenty-five years both the software industry and the research community have become increasingly aware of the difficulties associated with producing a high quality software requirements specification (van Lamsweerde 2000). It is widely acknowledged that requirements engineering process problems reduce the quality of software and undermine the effectiveness of the software development process (Lindland et al. 1994; Sommerville 2001). Indeed, my own previous collaborative work suggests that software organisations are very aware of the relationship between inadequate requirements engineering processes and, for example, high maintenance costs (Hall et al. 2001).

A great deal of excellent work continues to be done in developing ways to improve requirements processes. Much of this work focuses on the interface between developers and users. Novel approaches to eliciting and representing requirements have been developed alongside formal models of requirements engineering processes, for example, see (Lindland et al. 1994; Sharp 1994; Macaulay 1996; O'Neill et al. 1997; Sutcliffe et al. 1997; Loomes and Jones 1998; Gross and Yu 2001). Although these approaches often comment on the limitations of existing solutions, their methods tend to be independent of previous improvement models and processes.

1.2 The requirements engineering process defined

Terms such as 'requirements', 'specification', 'requirements engineering' and 'RE' are often used in the literature to embrace the whole of the requirements 'process' (Lindland et al. 1994). The term 'requirements engineering process' or 'RE process' as used in this study, refers to activities performed in the requirements phase that culminate in producing a document containing the software requirements specification (Jalote 1997). More specifically, the RE process is the set of activities required to gather, specify, validate and engineer a set of requirements (Britton 2000); (IEEE Software - Thayer and Dorfman 1990, page 1); whereas 'a requirement' is defined as "a feature or behaviour of the system that is desired by one or more stakeholders" (Britton 2000). This study focuses on the 'RE process' and not the individual feature or behaviour of the system.

My view of the RE process takes a complementary approach to existing work. I suggest that multiple factors affect the production of the requirements specification and an important class of factors are those internal to the development organisation. My approach is analogous to Procaccino et al's study (2002) of the multiple factors affecting software development success.

A glossary of acronyms and terms used in this thesis is given in Appendix A.

1.2.1 The requirements process is 'engineered'

Thayer and Dorfman (1990) consider the requirements process to be 'engineered' as practitioners need to select appropriate mechanisms to convert the elicitation, analysis, documentation and verification activities into a software requirements specification. RE was established as a separate field of investigation and practice in the mid-1970s (Loucopoulos and Karakostas 1995). The RE process includes both system requirements, i.e. the processes involved in understanding and analysing the problem originating from user needs; and software requirements, i.e. the processes required to produce a requirement specification that originate from the system requirements (Thayer and Dorfman 1990). The goal of RE is to determine a need and define the external behaviour of a solution (Davis 1994).

Therefore requirements engineering (RE) is a separate field of study as covered by many journal texts and conferences, and the 'requirements engineering process' relates to the many activities associated with the RE discipline.

1.3 The RE problem

Within the software community there is a common belief that RE is causing companies the greatest problem, see for example, (El Emam and Madhavji 1995a; Krasner 1997; Sommerville and Sawyer 1997; Leffingwell and Widrig 2000). It is widely accepted that the quality of the end product depends heavily on the accuracy of the requirements specification (Lindland et al. 1994). RE process problems are found to be persistent, pervasive and costly (Faulk 1990; Boehm et al. 1994).

There is a general consensus as to the 'types' of RE problems companies are experiencing (Isazadeh & Lamb, 1999; Patel, 1999, Curtis, 1988). However, most of the literature views the RE process in a piecemeal fashion when offering possible solutions; for example, different methods of tracing requirements, how to use scenarios in RE and how to capture and validate requirements in distributed systems as found in (IEEE 1999) and (IEEE 1997). Empirical research confirms that RE problems are indeed multifarious, yet inter-dependent, with each individual problem influencing project success (Hall et al 2002). Moreover, the impact of each individual problem is likely to differ in terms of severity. It would therefore follow that there is a need for an empirical evaluation as to how RE process problems are recognised, evaluated and prioritised. Further work is also needed in considering how the different solutions to RE process problems might be combined into one framework.

This research analyses individual RE process problems as a necessary first step to finding solutions. RE processes are identified and explored from a software engineering perspective. How software process improvement models, such as the SW-CMM (Paulk et al. 1995) and the CMMI (2001), address the RE process is also examined to explore where these models might be strengthened. Therefore the focus of this research is on RE process problem recognition, evaluation and assessment rather than on specific, tailored problem solutions.

1.4 RE process solutions

Researchers and practitioners understand that the earlier RE process problems or requirements defects are detected, the easier and cheaper they are to repair, as shown in the work of Boehm (1981). Lauesen and Vinter (2001) build on this observation noting that, "detection as well as prevention [of requirements defects] requires some effort in addition to usual development." This need for additional effort is also a major theme in the early empirical study of Bell and Thayer who state that, "the requirements for a system do not arise naturally; instead, they need to be engineered and have continuing review and revision" (Bell and Thayer 1976). The experience report of a high maturity organisation, the NASA space shuttle project, confirms that moving resources to the front end of software development can contribute to the reduction of delivered software defects (Krasner et al. 1994).

1.5 Modelling the RE process

I examine the RE process in order to model the factors that contribute towards the gathering, specifying, validating and engineering of a set of requirements. According to Panedo and Shu (1991) formalising the lifecycle process is key to software improvement. I view the RE process independently of specific 'software' lifecycle models. Although software lifecycle models can play an important role in software engineering as a general guide to software developers, they lack the detail required for an analysis of a specific process (Madhavji 1991; Penedo and Shu 1991). The RE process is therefore viewed in isolation, where

"*... abstractions and simplifications are made to enable the designer to conceptualise aspects of the problem in a holistic fashion, omitting reference to details and relationships that are not immediately pertinent to the task in hand*" (Loomes and Jones 1998, page 1).

Houdek and Pohl observe that RE activities such as elicitation and validation are heavily intertwined and question whether differentiation between these phases yields benefit (Houdek and Pohl 2000). Indeed, taking a purely process view of requirements breaks away from a partitioned lifecycle view. However, the R-CMM includes a traditional process view of the activities involved in the production of requirements as described in general software engineering texts such as Dorfman and Thayer (1997) and Pressman (2001). The five phases represented in the model (requirements management, elicitation, analysis, documentation and verification) bridge the gap between a conventional/structured 'lifecycle' view and a process view of RE. This traditional view has the advantage of using familiar vocabulary and creating an intellectual tool that allows the user to focus on different areas of the RE process.

1.5.1 Generic qualities of the model

The focus of this thesis is on company practices involved in developing bespoke software systems. It is necessary to make this distinction as the needs of companies involved in other forms of development such as 'commercial-off-the-shelf' development are likely to differ, as the role of the customer and system constraints have a different emphasis (Fairley and Rook 1990). Also, the more detailed and refined the model becomes, the less generic it is (Loomes and Jones 1998).

I view the RE process as integral to software development. Indeed, the resulting requirements specification may be revisited and changed throughout the project. Brooks captures the iterative nature of the RE process in his seminal paper where he states "..the most important function that the software builder performs for the client is the iterative extraction and refinement of the product requirements process" (Brooks 1987). However, in order to attempt to improve current representations of this front end of development the R-CMM captures only those activities that relate to the RE process.

The R-CMM aims to present complex activities in a way that can be easily understood. The SW-CMM provides a broad maturity framework which the R-CMM uses to describe these requirements activities. In developing the requirements process maturity framework, I follow set rules of model building as suggested in the literature (Rossi 1999; Koubarakis and Plexousakis 2002). A goal focus is implemented through a Goal/Question/Metric (GQM) approach as developed by Basili and Rombach (1988). The R-CMM includes project management processes, or 'organisational' processes that underpin the technical requirements processes. The decision to include these softer processes is supported by both my empirical work and the literature, where the lack of organisational control was found to be a major source of requirements process problems (Lubars et al. 1993; Hall et al. 2002b).

1.6 Project motivation

This study grew out of my collaborative work with the PPP project. An analysis of focus group interview data revealed many technical and organisational problems. In nearly all groups there was a recurring theme relating to problems associated with the RE process. Developers, project managers and senior managers all had an awareness of RE process issues that need to be addressed (Hall et al. 2002b).

An examination of the RE literature, together with my empirical study, highlights the need to make the requirements process problems that practitioners are experiencing more transparent and coherent. It could be that the approaches developed to address the problem of general software quality are not reaching the requirements phase of development. Alternatively it could be that quality frameworks are often presented as lists of desirable properties that do not give a systematic structure for achieving them or evaluating them (Lindland et al. 1994).

1.6.1 Software process improvement

The aim of SPI is to focus on improving the processes used to develop software in order to improve the quality of the product (Humphrey 1989). Some of the most popular SPI models are the Software Capability Maturity Model (SW-CMM) (Paulk et al. 1995) and the Capability Maturity Model Integration (CMMI 2001) both developed in the US by the Software Engineering Institute (SEI). Also, an emerging international standard for SPI is the SPICE or ISO/IEC 15504 model (SPICE 2003) that is planned for release in 2003/2004.

There is empirical evidence to show that SPI models can improve software quality (Herbsleb 1997; Cugola and Ghezzi 1998; Curtis 2000). The CMMI addresses many of the weaknesses of the SW-CMM by including more organisational processes associated with the requirements process. It also introduces a more flexible approach to process assessment. As the CMMI has only recently been released, it is difficult to assess its strengths and weaknesses. However, the RE process remains buried within its 700 plus pages of guidelines (Reifer 2000) and may therefore continue to be overlooked in the CMMI presentation. I explore whether the focus that the SW-CMM and the CMMI provide towards general software development can be adapted to frame the RE process in isolation.

A further motivation for creating the R-CMM is that organisations cannot always rely on external researchers or wait two years for the lengthy, external process assessment, to identify where their weaknesses are. Although companies can perform their own internal SW-CMM assessments, current methods combine the RE process improvement with the whole of software development. The R-CMM therefore includes an internal assessment component that allows practitioners to analyse their current RE activities with a view to prioritising where best to focus their improvement efforts.

1.6.2 Advantages of adapting the SW-CMM

I started to develop a SPI model that specialises in the requirements phase of software development to bring together key stakeholders involved in the requirements process and give them the opportunity to establish precisely where their problems lie. The R-CMM is sympathetic to the existing culture surrounding companies who are using the SW-CMM as their mechanism for SPI. Complementing this familiar model should help practitioners as they do not need to learn yet another improvement methodology. Applying the SW-CMM characteristic of capturing the 'repeatability' of best practices is particularly appropriate to the requirements process, as

" *'Requirements' in the wider sense are captured not only within specific projects, but also carry across between projects, embedded in cultures, organisations and communities* " (Loomes and Jones 1998, page 7).

I acknowledge that the SW-CMM is not a perfect model of SPI and give a critique of both its strengths and weaknesses in the literature review in the following chapter. However there are many reasons in favour of adapting the SW-CMM to create a specialised model of RE as the SW-CMM

- Contains guidelines for many requirements-related activities
- Is based on best practice derived from many years of empirical study
- Has a limited set of activities
- Is a known standard
- Has a proven record of achievement
- Is designed to be tailored to focus on specific process areas
- Continues to be supported by the SEI
- Has a maturity structure to help with process prioritisation
- Is goal focussed
- Integrates RE practices with software development

1.7 Complementary work

This is not the first attempt to represent the RE process in a maturity framework, for example, Sommerville and Sawyer (1997) also recognised this need. Their RE good practice guide, although based on maturity levels, does not link directly to existing maturity models. Other models of the RE process presented by requirements experts such as (Davis 1988; Thayer and Dorfman 1990; Jackson 1995b; Kotonya and Sommerville 1998; Arisholm and Sjoberg 2000; Leffingwell and Widrig 2000; Boehm 2001) all offer different approaches to solving the RE problem. Their properties for quality requirements, although using mainstream terminology, offer a confused message. Definitions of desirable qualities are often found to be vague, complicated and lacking in detail. Lists can be unstructured and overlapping, and some goals are unrealistic and even impossible to reach (Lindland et al. 1994). The practitioner is therefore left not only wondering which advice to take, but having opted for a method, may not be given the means to achieve their objectives.

1.8 Research aims

I concentrate on enabling practitioners to build a quality RE process that supports and integrates with software engineering activities. I aim to help practitioners to reach an understanding of how to tailor RE processes to meet their own needs, how to set realistic quality RE process goals and provide a means to achieve their goals. I do not merely present a list of useful RE process features that are independent of software, but provide a framework that guides users towards an integrated view of the RE process, where maturity goals are set to help with process prioritisation and implementation.

I aim to narrow the gap between RE process research (suggesting principles, techniques, languages and tools to help analysts understand a problem or describe a potential product's external behaviour) and the practice "where software customers understandably wonder if anyone is listening" (Davis 1994). This work pulls together disparate work in the field of software process improvement and RE process solutions in a way that is accessible to both practitioners and researchers.

1.9 Contribution to knowledge

I have built on the work of the SEI and the RE literature to create an outline model that combines technical RE processes together with supportive organisational processes. No previous work presents the RE process in terms of individual process capability that is governed by SW-CMM maturity goals.

I have combined individual RE solutions into one framework. The framework used represents an approach to software process improvement that is familiar to many practitioners. The model I develop, 'The R-CMM' is a unique, specialised, cohesive and comprehensive model that reflects RE key processes at incremental levels of capability.

My contribution to knowledge is a model that represents a new process view of the requirements phase. In this thesis I show how the model is built, where there are strengths in the model, and where possible improvements can be made to enable other researchers to build on my work and continue towards seeking methods to improve the RE process.

My hypothesis

A CMM-based RE process model can help to assess the maturity of the RE process.

1.10 Methodology

A major objective of this study is to try to develop a model that is relevant to the real problems companies are experiencing in their software development. For this reason I take an empirical line of enquiry where direct research is undertaken to learn how companies operate in practice. How RE is impacting software development within a diverse group of software companies is investigated using a combination of qualitative and quantitative approaches to data collection and analysis.

1.10.1 Focus groups

To gain a better understanding of the problems that practitioners are experiencing in their software development, focus groups were conducted involving over 200 practitioners. In these focus groups, researchers asked groups of software practitioners, "What are the problems and issues in software development in your company?" and "What are the obstacles to SPI in your company?" An initial content analysis of practitioner responses to these questions highlighted requirements as a major problem. Having established RE as a major issue, the focus group transcripts were revisited to investigate individual requirements problems and whether they reflect findings in the literature. Problem frequencies were placed into contingency tables to allow observations to be made. The result of this analysis is a general impression of the main problems practitioners were experiencing in software development and more specifically, in their RE process.

1.10.2 Model development

The next stage in my research involves building a model to support practitioners with their RE process. A rule-based procedure is followed where model development is initiated by creating and agreeing model criteria. Objectives are set to clarify the purpose of the model and outline what the model is expected to describe. These criteria steer development and are later used to help validate the model.

Model building activities involve abstracting characteristics and practices from three sources:

- SW-CMM architecture and RE processes
- best practices in the RE literature
- empirical findings

The RE and SW-CMM literature used in this study are based mainly on case studies, company experience reports and empirical evidence. This work provides a context for software process improvement and RE practices. The literature is used to gather recommendations put forward by a carefully selected group of experts in the fields of RE and SPI. However, the rationale for choosing processes to populate the model is based on addressing the RE process problems highlighted in my empirical study (Beecham et al. 2003d).

1.10.3 Model validation

Having created a model based on practitioner needs, a questionnaire is designed to validate how well the model meets the objectives of the study. Seven success criteria are externally assessed prior to proceeding further with model development. A group of experts in the field of RE and software process improvement looked at model components and responded to questions that directly relate to the success criteria. The results of this validation exercise highlighted areas in the model that should be retained, as well as areas that could be strengthened or clarified.

1.10.4 Summary of methods

I describe my empirical methods in sufficient detail to allow subsequent researchers to assess and replicate my work. The methods used explore how to create a RE process model that captures the needs of practitioners. The planning, implementation and reporting of the empirical research undertaken observe guidelines given by Kitchenham et al. (2002b).

1.11 Overview of thesis

This thesis is organized into four main sections that comprise nine chapters. The first part of the thesis gives a background to the work as given in this chapter, chapter two and three. Chapter two is a review of the related literature and chapter three describes the research methods used. Part two of the thesis is an empirical study of the problems in both software engineering in general and RE specifically as covered in chapters four and five. This investigation culminates in a proposal. The third part of the thesis presents a RE solution as developed in chapters six, seven and eight. These chapters describe the studies carried out to create and validate a model of the RE process based on SW-CMM architecture. Chapter nine is the fourth and final part of the thesis and presents conclusions and introduces future work.

PART 1: BACKGROUND

1.11.1 Chapter one: Introduction

1.11.2 Chapter two: Literature review

Chapter two presents examples of how the literature views RE problems and their solutions. As a complex area of software development, the literature is reviewed in order to place the RE process in the context of broader software engineering activities. The literature indicates that a solution to general software development problems may be found in the Software Process Improvement (SPI) methods. The SPI literature is therefore also reviewed in order to uncover the strengths and weaknesses in the SW-CMM as a model of SPI. A review of SW-CMM literature is presented to indicate how the model supports the RE process. Finally, a background to some principles of model development is given to guide the work in this thesis.

The themes of this chapter are:

- The impact that RE has on software development.
- Requirements process problems and solutions.
- Methods used to assess the strengths and weaknesses in software processes.
- How the SW-CMM, as the most applied model of Software Process Improvement, is helping to support the RE process.
- Principles of building a 'best practice' model to support process improvement.

1.11.3 Chapter three: Methodology

Chapter three presents the design of the research process. It describes the approach adopted in this research and the particular research methods and techniques used. It explains the rationale behind choosing the research design and shows how qualitative and quantitative methods are used in data collection and analysis.

PART TWO: EMPIRICAL STUDIES INTO RE PROBLEMS

1.11.4 Chapter four: Software development problems: An empirical study

Chapter four is an empirical investigation into whether RE-problems highlighted in the literature are reflected in the companies in my study. In order to either refute or confirm the literature I examine the focus group transcripts from previous work on SPI and identify common problems. Groups of software development problems are identified through a grounded theory and content analysis approach (Glaser and Strauss 1967; Krippendorff 1980). The reliability of these classifications is confirmed through the use of a Cohen Kappa inter-rater test (Dunn 1989). Using a descriptive statistical technique, Correspondence Analysis (Greenacre and Blasius 1994), I give a graphical representation of how each practitioner group views software development problems. This chapter shows that RE process problems are causing practitioners problems that are likely to impact the whole of development. I am also able to establish that, in my sample, there is a movement away from RE process problems as organisations mature to higher SW-CMM levels. This suggests that some of the SW-CMM solutions are indeed working. The work presented in this chapter is given in (Beecham et al. 2003d).

1.11.5 Chapter five: RE process problems: An empirical study

Chapter five re-visits the PPP transcript data with a view to making a more detailed analysis of RE problems experienced by practitioners in the study. Through the use of content analysis, two types of problems are identified: RE process/technical problems, and RE organisational /management problems. This work guides me to break down the RE process into sub-processes that include stakeholder communication, skills audit, training and resource allocation. This finer-grained analysis is used to guide the development of the RE process model. The study is generated from collaborative work as discussed in (Hall et al. 2002a; Hall et al. 2002b).

PART THREE: RE SOLUTIONS

1.11.6 Chapter six: Models used to support process improvement – building a framework

In chapter six I build a framework for the RE process improvement model. I look to the literature for methods to support quality improvement and note reasons for and against using the SW-CMM as developed by the SEI (Humphrey 1989; Paulk et al. 1995). I also review the work of Basili and Rombach (1988) in order to consider their Goal Question Metric (GQM) paradigm as a method for presenting processes that are goal and measurement based. In this chapter I note the objectives for building the model, and list my success criteria. Using my gathered evidence and knowledge, I create a framework in which I represent RE best practices. This study is based on work presented in Beecham et al. (2003b).

1.11.7 Chapter seven: Requirements engineering best practices

In chapter seven I present candidate processes for a baseline RE process improvement model and give my rationale for defining and populating the R-CMM. Three sources are used to create a bank of RE best practices to populate the R-CMM. These sources are:

- Empirical findings:
  As the aim of the R-CMM is to support the needs of the software community, I limit my search to solutions that solve problems highlighted in the empirical studies (Hall et al. 2002a; Hall et al. 2002b; Beecham et al. 2003d).

- RE Literature
  The RE literature is sourced to provide solutions to problems identified in the empirical studies. In an attempt to create a model with a wide application and rigour, multiple sources underpin each individual solution. I thereby avoid basing solutions on single case studies or text book recommendations that lack an empirical dimension (Bach 1999).

- SW-CMM
  Where possible, I source SW-CMM best practices as I have the framework in place and endeavour to 're-use' given examples rather than create my own (Wiegers 1998a; Wiegers 1998b; Humphrey 2002).

This chapter is based on work presented in Beecham et al. (2003c).

1.11.8 Chapter eight: Validating the Requirements Capability Maturity Model (R-CMM)

Having built several model components, chapter eight explains how a cohesive segment of the R-CMM is presented to an external group of experts to validate. I use methods drawn from the literature to substantiate my validation methods. A questionnaire is designed specifically to validate the R-CMM. Validation techniques include the application of confidence levels and inter-rater testing as well as nonparametric significance tests such as Pearson chi-squared test, Mann-Whitney U and McNemar. Results from this study enabled me to gain a more objective view of how closely my model meets my original objectives and success criteria (Beecham et al. 2003a).

The validation and evaluation of the model is defined as follows. Validation is the substantiation that the components within the model's domain of application possess a satisfactory range of accuracy consistent with the intended application of the model (Sargent 2000). Carson and Robinson's definitions are pertinent to this study where validation is defined as "the process of ensuring that the model is sufficiently accurate for the purpose at hand" (Carson 1986), or whether the right model is being built (Robinson 1997).

Verification is defined as the process of ensuring that the model design (conceptual model) has been transformed ... with sufficient accuracy (Davis 1992), testing whether the model is built correctly (Robinson 1997), and ensuring that the model components are correct (Sargent 2000). It is not possible to verify that the model is built correctly at this at this stage of development as this would require the model elements to be implemented.

Evaluation, however, encompasses both validation and verification activities along with the model's quality, usability and utility assessment (Gass 1983). At the end of the chapter I evaluate how well the model has been validated taking a broader view of the work.

PART FOUR: CONCLUSION

1.11.9 Chapter nine: Summary and conclusion

Chapter nine presents a summary of this research programme and explains how the R-CMM helps to provide a better understanding of the RE process. This concluding chapter includes a critique of the overall strategy and how, in hindsight, it might be improved. This chapter also reflects on the research methodology both in terms of its success and how it might be used in future research. Finally, suggestions are made as to how other research might evolve from the work presented in this thesis.

1.12 Relationship between this research and the PPP project

The EPSRC-funded PPP project was established to investigate empirically how the software industry is approaching Software Process Improvement (SPI) with a focus on the impact people have on the product and process. An initial analysis of the PPP data identified problems in the RE process. My involvement in the project began in October 2000 when I was recruited to look at the RE process area of software improvement in more detail. Initially I classified problems raised by PPP data that confirmed the RE process as a major problem for all the software development organisations in the study.

Some of the data collection processes described in chapter three and in the study chapters were undertaken in conjunction with the overall PPP project, whilst others were specifically undertaken for this research. This is explained as follows:

- The data collection process for the study of general software development problems was a by-product of PPP focus group discussions. I analysed and used the data previously collected by the PPP team.

- The data collection process for the study of specific RE process problems was a by-product of PPP focus group discussions. I analysed and used the data previously collected by the PPP team.

- The data collection process for the validation of the RE process improvement model was undertaken specifically for this research.

All the data analyses reported in this thesis were conducted specifically for this research. The findings of the thesis derive from these analyses.

Chapter Two: The RE Process and the Software CMM

"A dwarf standing on the shoulders of a giant may see farther than a giant himself"
(Democritus to the Reader in Anatomy of Melancholy by Burton 1621)

2.1 Introduction

This chapter presents a review of the literature for this research. Sources are drawn from empirical studies, case studies, experience reports, recognised standards and texts. Multiple sources are used "as there must be no deference in the scrutiny" (Shipman 1997, page 5).

The work of experts is distilled to provide a background to RE problems and how they might be resolved. Identifying problems is an essential element in software process improvement (SPI). Sommerville and Sawyer (1997), for example, state that the first question that a company should ask is: "What are the problems with our current processes?" El Emam et al (1996) believe "it is important to understand the enabling and the inhibiting factors involved in SPI, particularly those that management can control". The literature indicates that a solution to general software development problems may be found in the Software Process Improvement (SPI) methods. The SPI literature is therefore also reviewed in order to uncover the strengths and weaknesses in the SW-CMM as a model of SPI.

2.1.1 Objectives of this literature review

The aim of this chapter is to provide background material that puts this research into context. It sets the scene for the contribution this research will make to existing knowledge. The literature is reviewed to gain a balanced understanding of the following:

- The impact RE has on software development.
- RE process problems and solutions.
- Methods used to assess the strengths and weaknesses in software processes.

- How the SW-CMM, as the most applied model of Software Process Improvement, is helping to support the RE process.
- Principles of building a best practice model to support process improvement.

The rest of this chapter is structured as follows:

The next section provides a background to the requirements process as an engineering discipline and gives examples of how requirements processes feature in software development. This is followed in section 2.3 by an overview of requirements engineering problems as reported in the literature to include both technical and organisational issues. Section 2.4 presents some best practices and models that are providing solutions to the requirements problem. In section 2.5 the Software Process Improvement (SPI) concept is introduced along with some of the models that support this approach. Section 2.6 takes a detailed look at how the SW-CMM supports the RE process. General principles of model building are then explored in section 2.7. These principles act as a guide to the proposed development of a specialised RE process improvement model. This chapter is concluded in section 2.8 with a summary of the main findings highlighted in this literature review.

2.2 The RE process

This section gives a background to the RE process and shows the impact RE has on software development. The literature presents a rich catalogue of RE process problems and those that create a recurring theme are given here. Solutions to these problems are then presented to give an indication of how research is supporting the RE process.

2.2.1 The RE process in context with software engineering

Software engineering texts are burgeoning with references alluding to the importance of the RE process, e.g. (Boehm 1981; Dorfman and Thayer 1997; Jalote 1997; Sommerville 2001). There are two main reasons for the weight of research in this area. Firstly, the RE process is highly complex with numerous ways of approaching the task; and secondly, mistakes made within this front end of development can be extremely costly in terms of an organisation's resources and reputation. Boehm estimated that late correction of requirements errors can cost up to 200 times more than corrections made early in the development cycle (Boehm 1981). In his well-cited paper, Brooks (1987) singles out the RE process as the most difficult and important phase in software development:

"The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later" (Brooks 1987, page 17).

Researchers and practitioners have been aware of the RE burden for at least 25 years. Arguably one of the earliest recognitions of the impact poor quality requirements have on software development is noted in the empirical study of Bell and Thayer (1976) where they conclude that requirements do not arise naturally but need to be engineered, continually reviewed and revised. Yet improving the quality of requirements is hindered by the fact that the RE process is less understood than other software development area (Smith 1998).

Poor quality requirements continue to place a strain on development suggesting that methods for capturing and supporting the RE process are not keeping pace with the ever growing demands for more complex and sophisticated systems. The scale of the problem is observed in a survey of over 8000 projects in the US where poor quality requirements were the main cause of one third of the projects not being completed (Standish Group 1995). This problem is replicated in a European survey aimed at highlighting problematic areas in software development. This ESPITI project, that covered 17 countries and had 3,401 responses, found that producing the requirements specification and managing customer requirements are causing software organisations the greatest difficulties (Ibanez and Rempp 1996).

2.2.2 Market driven and customer specific requirements

Sommerville (2001) views software engineering as addressing two types of software production; 'bespoke products' (or customer specific, unique products), and 'generic products' (or market driven, commercial off the shelf (COTS) products). Jackson (1995a) acknowledges that the complex activities associated with RE are more likely to apply to bespoke software system development than COTS development. This is mainly due to the need to capture and engineer the particular needs of the customer in bespoke systems development. This disparity in approach is supported by the empirical study of Lubars et al (1993) who observe that software development projects that lack a readily identifiable customer take a less structured approach to development. As Loucopoulos and Karakostas (1995) and (Potts 1995) conclude, market driven and customer specific requirements have different characteristics and are therefore often treated differently within a development process.

As I am interested in modelling the key processes involved in producing a requirements specification, this study will concentrate on the needs of bespoke system development as defined within RE.

2.2.3 The RE 'process' defined

This section explains how RE as a separate software engineering discipline is viewed as a 'process', where the different activities that lead to the production of the software specification are explored.

"Traditionally, RE was seen as an early phase in the system development process. As proposed in the mid 1990s, shorter time to market, technology changes, and frequently changing environments force a shift in this traditional view. So, RE should be understood as a continuous activity that manages requirements evolution through-out the system life cycle and between system boundaries" (Dubois and Pohl 2003, page 14).

In a recent paper, Pinheiro (2003) states that elicitation, analysis, and validation are at the heart of the RE process. Pinheiro continues that the careful process of studying, understanding, and analysing requirements is necessary to deal with the complexities of requirements elicitation. Validation is also essential because "if we do not know whether we have the right requirements, then we also do not know if software built to meet these requirements will fulfil its objectives" (page 184). Also, as Cottengin (2002) explains, viewing the RE process in cohesive phases such as elicitation, analysis and validation helps to identify where system development problems arise as

"... the seeds of system failure are often sown ... in the requirements elicitation process. Many organizations lack the ability to consolidate and reconcile multiple stakeholder viewpoints" (Cottengim 2002, page 26).

Despite Pinheiro and Cottengin's recent papers projecting the traditional view of RE as depicted in texts such as (Loucopoulos and Karakostas 1995; Dorfman and Thayer 1997; Jalote 1997; Kotonya and Sommerville 1998; Pressman 2001; Sommerville 2001), some requirements experts, such as (Dubois and Pohl 2003) question whether this 'phased' view is helpful or indeed reflective of the activities involved in RE. For example, Nguyen and Swatman (2003) observed that opportunistic RE methods (that are believed to reflect the true activities in creative RE) do not follow the cyclical, or sequential pattern of development presented in the traditional RE texts.

Andreou (2003) also questions the phased view of RE noting that the explosion in telecommunications and continuous growth of the Internet has caused significant revisions in certain phases of the lifecycle models. This is mainly because the knowledge of the system and an understanding of requirements in agile methods are not all learned early in development. For example, the need for continuous change in content and functionality in web applications forced a quicker development of software products. A major shift in the relationship between customer and developer is that users are no longer a passive component of the overall system that is subjected to 'interrogation' by analysts to reveal the right needs (Andreou 2003). However, the main difference between agile methods and phased RE methods is not whether to do RE but when to do it (Kovitz 2003).

Taking an opposing view, some of the advantages of a phased view of the RE process are explained by Hofmann and Lehner (2001), who also emphasise the importance of the role of the stakeholder in RE. They state that typically requirements are first elicited, and then 'modelled' to specify a solution. Modelling describes a perceived solution in the context of an application domain using informal, semi-formal, or formal notations. The gradual normalization of such models in terms of the requirements leads to a satisfactory candidate specification, which must then be validated and verified. This gives stakeholders feedback on the interpretation of their requirements so they can correct misunderstandings as early as possible.

The many lifecycle models and project management methods that have been developed to support software development all include RE as an integral part of development. In practice therefore, the RE process is shown to take many paths, and as Potts (1993) points out more work is needed on systematic methods for requirements elicitation or definition. Despite the proliferation of process models however, field studies show that very few organisations explicitly define or tailor their RE process (Hofmann and Lehner 2001).

2.3 RE problems

"Although software engineers and managers often know their problems in great detail, they may disagree on which improvements are most important" (Paulk et al, 1995, page 10).

The problems encountered in the RE process as recorded in the literature are not new. Bell and Thayer observed in their 1976 empirical study that inadequate, inconsistent, incomplete and ambiguous requirements are numerous and have a critical impact on the resulting software. Looking specifically at the RE output, Meyer (1985) noted that a specification that does not reflect real needs, and that is incomplete, contradictory and ambiguous may have a disastrous effect on subsequent development steps. However, as Lindland et al (1994) point out, having a goal of a complete requirements specification may be unrealistic. Yet an experience report shows that to improve the quality of requirements, characteristics such as correctness, consistency and ambiguity need to be evaluated (Smith 1998) as based on IEEE standard 830-1993 (IEEE 1994).

Curtis et al (1988) adopted an empirical perspective to reveal that RE issues were a 'recurring theme' in interviews with practitioners who cited three problem areas:

1. how system requirements were understood;
2. how their instability affected design;
3. how requirements were communicated throughout the project.

These problems have been echoed throughout studies in subsequent years. Bach (1995) confirms that difficulties remain in understanding requirements and emphasises the need for appropriate skills, experience and methods. Faulk (1997) notes that people often do not know what they want and places responsibility with the software developers (Faulk 1990). Patel (1999) focuses on the impact of requirements change and Donzelli and Iazeolla (2001) provide empirical evidence of the impact requirements instability has on effort and delivery times. Macaulay (1999) describes continual problems with internal and external communication and discusses requirements as a group activity.

El Emam and Birk (2000), in their validation of the ISO/IEC 15504 measure of software requirement analysis, report that 80% of Management Information Systems, 70% of military and 45% of contract or outsourced projects are at risk of creeping user requirements. The two areas in their study that gave practitioners the most concern being, producing the requirements specification, and managing customer requirements.

2.3.1 Organisational problems that impact the RE process

The RE literature recognises the importance of organisational processes in software development. The work of (Lubars et al. 1993) showed that organisational issues caused practitioners more problems than technical issues. And Cottengim (2002) notes that all too often new requirements gathering methodologies are tried without the attendant examination of the organisation's underlying characteristics. El Emam and Madhavji (1996a), in their RE process improvement empirical study, found that it is the organisational dimension of a process maturity model that influences the quality of RE.

Another body of work suggests that organisational factors that support RE are often weaker than technical factors (Humphrey 1989; Lubars et al. 1993; Perry et al. 1994; El Emam and Madhavji 1995a; Fox and Frakes 1997; Glass et al. 2002). There appears to be a general weakness in SPI support where organisational aspects are overshadowed by the many research efforts aimed at developing technology support (Rossi 1999). According to Humphrey, when asked to name their key problems, few software professionals even mention 'technology', their main concerns being open-ended requirements, uncontrolled change, arbitrary schedules, insufficient test time, inadequate training and unmanaged system standards (Humphrey 1989). While Boehm (1981) found that it was only when a system was structured from both a human and technical perspective that a well-balanced system resulted satisfying all operational goals.

Herbsleb and Goldenson (1996) in their field study of CMM experience also found organisational issues to be the major impediments to successful process improvement. Either organisations are not aware of the problems organisational issues are causing, or are unable to manage this softer side of software development. Looking at specific organisational problems Hofmann and Lehner (2001) found that a lack of training led to teams that were less familiar with the RE process. While Humphrey (2002), in his section about managing teams in process improvement, states that

"*... the biggest single problem ... is training. With few exceptions, managers want the benefits ... but are reluctant to invest in the required training*" (Humphrey 2002, page 63).

A major problem identified in the literature relates to identifying and involving stakeholders. A survey carried out by Boehm and his team showed that practitioners' major concerns with their organisation's typical RE approach included the lack of key stakeholder involvement (Boehm 2001). The Standish Group's Chaos report

(1995) also identified 'lack of user input' as contributing to 12.8% of project failure. Further literature identifying the lack of stakeholder participation as a major issue in RE include the work of Hofmann and Lehner (2001) and El Emam et al. (1996).

2.4 Solutions to diverse RE problems

"We expect methods to be panaceas – medicines that cure all diseases. This cannot be" (Jackson 1995, page 4).

The proliferation of publications, text books and conferences that focus on RE confirm the importance, diversity and complexity of this process. RE deals with domains such as banking, transportation and manufacturing and tackles tasks such as administrative support, decision support and process control. And it addresses environments such as human organisations and physical phenomena. It is therefore perhaps helpful to view RE as covering multiple intertwined activities (van Lamsweerde 2000; Procaccino et al. 2002). As a result, however, it is difficult to find universal solutions that apply to reactive or customer specific systems, which is why so much research is based on a specific focus and dedicated techniques.

A model that takes a holistic view of RE process improvement is found in the work of Sawyer et al. (1997), and Sommerville and Sawyer (1997) with their development of a RE good practice guide. Retaining a balance between technical and organisational support is explicitly noted in the work of Dobson and Strens (1994) who conclude that technical and organisational issues are so interrelated that optimising one group alone is insufficient as the unsupported process is likely to have an adverse effect on both areas.

Fordham (1999) alleges that although technical issues are important, it is the relationship between technical efficiency and social considerations that is paramount to the success of any business. Fordham advocates creating

"... a blend of technology, people and process to provide a balanced score card of activities that can address our goals more successfully – people overlooked or misused is the most critical resource in the equation" (Fordham 1999, page 611).

The literature provides many solutions and counter examples that address problems in RE. There appears to be a problem with relating theory to practice, as so often the supporting processes are overlooked. Fenton (2001) argues that recent progress made in the empirical software engineering field has failed to impact mainstream practice. However, Glass speaks up for practitioners, noting that research is not recognising the good practice that is being realised in the high quality software being produced (Glass 1996; Glass 1997). More specifically, Berry and Lawrence argue that the gulf between software engineering research and practice is no more evident than in the field of RE (Berry and Lawrence 1998).

A European survey found that adoption levels of RE practices are very low (Dutta and van Wassenhove 1997). Procedures for ensuring appropriate levels of user/customer/marketing input; for controlling changes to requirements; designs and documentation; and prototyping for validating requirements all have a less than 60% take up. The most extreme finding is that tools for requirements traceability have a 22% adoption (Dutta and van Wassenhove 1997).

The solutions below are necessarily generic and are divided into organisational RE processes and technical RE solutions.

2.4.1 Organisational RE process solutions

Non-technical difficulties in software engineering are repeatedly recognised in the literature with the introduction of new methods of support. For example Christie (1999) proposes a tool to simulate the complex behaviour of processes that involve creative and human-intensive activities; while Pfleeger and Rombach (1994) give several illustrations of how measurement based research techniques and development tools can support the management of software improvement in a special issue of IEEE Software. Cugola and Ghezzi (1998) investigate automated RE support. Fayed (1997) notes some organisational and technical reasons why organisations are opposed to process improvement, concluding that managing software processes is a 'necessary evil'.

The organisational processes listed below relate to the management of RE activities.

- Establish RE process and document

A well defined RE process leads to a flexible system that is quick to respond to change (e.g. links to resources, traceability, and is cohesive).

"To succeed you must integrate your technical, cognitive, social and organizational processes to suit your project's particular needs and characteristics" (Hofmann and Lehner, 2001, page 66).

Also, clearly documenting the business and overall Management Information Systems strategies that align to missions, goals and priorities is recommended (El Emam and Madhavji 1996a). The maturity of this process was found to have an effect on the requirements management.

"One of the most common reasons systems fail is because the definition of system requirements is bad" (Scharer 1990). According to Rule (2001) and Paulk *et al.* (1995), one methodology should be used project wide, e.g. waterfall, spiral, rapid and joint application development, extreme programming or rational unified process. Further references in support of establishing agreements and clear definitions of the process are (Sommerville and Sawyer, 1997; Cugola and Ghezzi, 1998; Sawyer et al., 1997; Pfleeger and Rombach, 1994; Fayad, 1997 and Christie, 1999).

- Establish responsibilities in RE

The organisation should establish project responsibility for analysing the system requirements and allocating them to hardware, software, and other system components (Paulk et al, 1995). The importance of taking responsibility for the processes involved in RE is further confirmed in McFeeley (1996).

- Recognise training needs in RE

A training programme should be implemented to recognise and meet technical and organisational RE needs within the project (Paulk et al, 1995).

Guides to incorporating training within projects is also given in ami (1992) where the emphasis is on 'properly administered' training, stating that "an assessment of the different needs and levels of training has to be made" (page 24). Humphrey (2002) adds to this discussion, stating that training is required to create a cohesive team that has a common understanding of the purposes and methods used in software improvement.

- Identify and involve stakeholders in RE

Stakeholder identification is central to Sommerville and Sawyer's (1997) RE model where: "The stakeholders in a system should always be explicitly identified in the RE document" (page 73). Dorfman (1990) states that good requirements should include an

"agreement among developers, customers, and users on the job to be done and the acceptance criteria for the delivered system" (Dorfman 1990, page 4).

Stakeholders involved early in the project increase the understanding of the RE process being used; and requirements prioritized by stakeholders drive successful RE teams (Hofmann and Lehner 2001). There is a need to develop a trust and a shared vision of what the project is trying to achieve with users who are part of the system. It is therefore necessary that users' capabilities are explicitly grown with the system for continued involvement, expression, participation and commitment (Middleton and McCollum 2001). Cottengin (2002) notes that the lack of ability to consolidate and reconcile multiple stakeholder viewpoints in the requirements elicitation process is the start of many software development problems.

Users should always participate in the RE process (El Emam and Madhavji 1995a). Although management commitment and support is needed from all levels of the company (Diaz and Sligo 1997; Mellis 1998; Willis et al. 1998; Ahuja 1999; Pitterman 2000), the buy-in of the technical community is also necessary (Herbsleb et al. 1994; Herbsleb and Goldenson 1996; Mellis 1998; Dybå 2000).

- Allocate resources to RE

Paulk et al (1995) recommend that organisations provide adequate resources and funding for managing the allocated requirements in the project. The RE process is a microcosm of the software process and as such organisations need to directly support RE activities by building an understanding and an awareness of the costs and benefits and commiting the resources necessary (McFeeley 1996).

Not only does the RE process need resources to perform the activities, part of its activities is to provide a good basis for resource estimation (cost, personnel quality and skills, equipment and time) (Doftman 1990). El Emam and Madhavji (1996a) found that integrating the budgeting process with business priorities had a positive effect on the quality of the RE process.

- Understand skills needed in RE

Hofman and Lehner (2001) suggest that successful RE is dependent on matching the needs of the project to the skills of the personnel. Establishing a process to identify skills needs within the project (for example, the skills needed in requirements elicitation) is a recommendation in Curtis et al's (1995) People Capability Maturity Model. There is a general discussion about personnel and the sensitive issue of how to rate personnel capability and personnel experience in (Boehm 1981).

El Emam and Madhavji (1995a), in their field study of RE practices, dedicate a section to skills. They recommend that appropriately skilled people be assigned to analyst and architect positions. They note the importance of identifying and involving skilled users in the RE process especially the principal user - project managers should also have a high capability in the RE phase.

- Promote stability in the project

Having a disciplined software engineering process will help address many 'accidental' difficulties in software requirements (Faulk 1990). Weinberg (1998) takes up this theme stating that

"To achieve a stable project over a long period of time, a manager must encourage the project to function .. with a fresh supply of trainees coming in one end and a stream of experienced leaders coming out of the other" ... and warns that "*A project is not a house of cards which collapses when a single key person is removed .. when management thinks it is, the prophecy becomes self-fulfilling"*. Finally, *"if a [practitioner] is indispensable, get rid of him as quickly as possible!!"*
From chapter six, 'Stability through change' pages 96-99 in Weinberg (1998).

A way to create a stable environment that will support the RE process is found in (McFeeley 1996), where McFeeley advocates that organisations establish a software process improvement infrastructure in order to build the mechanisms necessary to help the organisation institutionalize continuous process improvement. The stable environment achieved through SPI methods will have a positive effect on the RE process.

- Relate RE processes to business goals

Research indicates that if RE process improvement initiatives are focussed on current business needs and are understood and agreed by management they are more likely to be implemented (McFeeley 1996). This goal focus is further advocated by (Rifkin 2001) and (Potter and Sakry 2001) in their work on software process improvement methods. Fayad (1997) also supports this goal approach to process improvement explaining that "software development organizations exist to develop software rather than processes" (page 103).

The process of setting realistic RE goals is important for both

1. achieving an acceptable level of improvement for the RE phase to solve recognised problems, and
2. setting functional and non-functional 'requirements'.

As Linberg points out,

"when there is a perception that the requirements are unrealistic, software developers may become discouraged and not fully commit to the goals of the project" (Linberg 1999, page 178).

Studies show that all people involved in software processes must be actively committed to their company's improvement goals and must be included in the practical implementation of processes (Diaz and Sligo 1997; Krasner 1997; Baddoo 2002). Stelzer and Mellis (1998) warn that unless companies openly involve staff at all stages during implementation of improvement programmes, investment and best efforts are wasted. Indeed, Horvat et al (2000) suggest that the success of SPI projects depends on the acceptance of its goals and tasks by every single employee.

Having 'set' the goals, each goal needs to be monitored. Solingen and Berghout (1999) suggest that goals are reviewed and approved by a project team before data collection can actually begin. The review session should focus on:

- Do project members agree upon the defined goals, questions and metrics?
- Do project members identify any missing or unnecessary definitions?

2.4.2 Technical RE process solutions

- Prioritise requirements

Successful RE teams manage requirements priorities 'to specify prioritized requirements, the RE team develops various models together with prototypes'

(Hofmann and Lehner 2001). McFeeley has a section dedicated to prioritizing activities and developing an improvement agenda (McFeeley 1996).

- Control changing requirements (to include requirements creep/growth)

Leffingwell and Widrigg (2000) recommend actively managing changing requirements to keep the project under control and help ensure the reliable, repeatable production of high-quality software products. A strong requirements traceability process may help to control requirements growth, however it is important to use the correct traceability method (Knethen et al. 2002). Motorola have identified that effective change communications, configuration management and control over unauthorized changes help to manage changing requirements (Smith 1998).

Specific methods recommended to counter the problem of changing requirements involve recognising and anticipating volatile requirements such as mutable requirements; emergent requirements; consequential requirements and compatibility requirements (see (Kotonya and Sommerville 1998) p.116).

- Recognise and work with vague requirements

Vague requirements or 'requirements uncertainty' are found in requirement documentation that is incomplete and flawed (El Emam and Madhavji 1995a; Moynihan 2000). "The whole purpose of the requirements process is to reduce ambiguity in the development process" (Gause and Weinberg 1989) page 217.

El Emam and Madhavji note that the greater the requirements uncertainty the greater the amount of changes to the RE documentation (El Emam and Madhavji 1995a). To solve this problem involves recognising the skill levels required in developers and users and assigning the necessary skills to the project. The work of El Emam and Madhavji clearly shows how inter-dependent the RE process is, with requirements uncertainty depending on skills management and effecting requirements changes.

Davis et al. list 'unambiguous' requirements specified in the software requirements specification (SRS) on the top of their requirements quality list, and state "an SRS is unambiguous if and only if every requirement stated therein has only one possible interpretation" (Davis et al. 1993, page 142). Davis et al. dedicate a section in their seminal paper to unambiguous and complete requirements and suggest ways these may be measured and controlled.

- Promote user understanding

A precise and richly detailed understanding of expected behaviour is needed to create effective designs and develop correct code (Faulk 1990). Scharer (1990) explains that users have different goals and approaches to requirements from system analysts. Scharer suggests that although users provide the system definition, the systems people are responsible for it, and that if users can be made to understand their own needs then comprehension is positively affected.

- Implement traceability method

Creating a link or definable relationship between entities is recommended by (Watkins and Neal 1994) as "You can't manage what you can't trace". Successful RE teams maintain a requirements traceability matrix to track a requirement from its origin through its specification to its implementation (Hofmann and Lehner 2001). Implementing a workable requirements traceability method will help prevent losing work and will promote sharing work across teams.

The literature is rich in examples of traceability methods to suit different needs, to include requirements re-cycling and legacy systems (Sutcliffe et al. 1999; Leffingwell and Widrig 2000; Hofmann and Lehner 2001; Knethen et al. 2002).

- Assess feasibility and risk in project

Assessing the feasibility of a project should include risk assessment, where software risks associated with cost, resource, schedule, and technical aspects of the project are tracked (Paulk et al 1995). Analysts may need to steer the client away from requirements that cannot be met within the budget and schedule constraints (Coad and Yourdon 1990). Boehm's spiral model of software development has estimating risk in a software project as a central theme (Boehm 1988).

The success of projects requires that system boundaries are defined (Sommerville & Sawyer, 1997). Curtis et al (1988) found that accurate problem domain knowledge is critical to the success of the projects. Finally, Patel (1999) suggests that risk can be minimised through object oriented technology that allows both global and local aspects of requirements to be captured.

- Manage complex requirements

Large-scale projects can span many years and different sites, and can be highly complex. They may need to be highly reliable, safety critical and customized (Shere 1988). Object oriented analysis is a method designed to manage complex requirements through principles of abstraction, information hiding, inheritance and methods of organisation (Coad and Yourdon 1990 page 275) and (Fayad 1997).

Leffingwell and Widrig (2000) recommend that complex systems have a requirements specification for each sub-system and non-trivial application. In these cases, requirements must be captured and recorded in a document database, model or tool.

Techniques such as functional decomposition and input-output analysis reduce complex systems into manageable subsystems but may not help with complex organisational issues (Yu and Mylopoulos 1997). The i* framework is a method that identifies enterprise integration solutions for complex technical and human organisational environments (Yu and Mylopoulos 1997).

- Validate RE activities

Validation of all RE activities will help to strengthen the process. Failure to do so will allow poor practices to continue, as McFeeley points out:

"*People typically repeat past behaviors, including those that lead to success and those that do not. The organization must ensure that mistakes are not repeated that may have caused similar initiatives to fail in the past*" (McFeeley 1996, page 94).

According to Basili (1995b) "Any engineering process requires feedback and evaluation." Taking a process view of RE therefore means that measurement is an ideal mechanism for feedback and evaluation.

"The measurements and information fed back to developers, managers, customers and the [organisation] help in the understanding and control of the software processes and products and the relationships between them" (Basili 1995b, page 23).

Referring to the requirements document itself, best practice shows that successful teams repeatedly validate and verify requirements with multiple stakeholders. They use peer reviews, scenarios, and walk-throughs to improve the specification throughout the software's life cycle (Hofmann and Lehner 2001).

(Davis et al. 1993) also stress the importance of validation and suggest that finite, cost effective techniques that can be used to verify that every requirement is satisfied by the system as built. Davis et al. continue by noting the difficulties attached to verification and suggests methods for controlling difficult requirements. Technical reviews and inspections by trained personnel prove their value by a high software requirements specification defect removal efficiency (Smith 1998).

2.4.3 Solution overview

According to Davis and Hickey (2002) effective RE requires utilising knowledge to synthesise effective solutions. A traditional view of RE requires developers to possess the following

(1) knowledge of the problem domain

(2) knowledge of existing solutions within the solution domain, and

(3) knowledge of processes, methods and tools used in the practice of RE

And only recently, has research begun to recognise the need for a fourth area of knowledge:

(4) knowledge of how to decide which processes, methods and tools make most sense as a function of certain aspects of the problem domain, the specific problem being addressed, the people involved, and so on (Davis and Hickey 2002).

This fourth knowledge requirement suggests that RE engineers need to take a holistic view of the RE process in order to assess which of the many solutions offered in the literature is appropriate for their specific needs. They need a means by which they can decide, debate and assess how best to utilise their knowledge.

The literature confirms that 'the appropriate mechanisms' required to facilitate RE activities incorporate both technical and organisational processes (Thayer and Dorfman 1990). Van Lamsweerde (2000) points out that much of the RE literature is devoted to techniques for modelling and specification. Although there is certainly a need for both the technical and organisational RE support, this thesis focuses mainly on the mechanisms and management of the RE process. This process management, however, depends on the excellent work that continues to be done with introducing new approaches to eliciting and representing requirements alongside formal models of RE processes, see for example the work of (Sharp 1994; Macaulay 1996; O'Neill et al. 1997; Sutcliffe et al. 1997; Andreou 2003).

2.4.4 Modelling RE solutions

Software researchers and engineers continue to seek ways to improve their ability to build software. These methods include

- Structured design and programming
- Abstract data types
- Object-oriented design and programming
- CASE tools
- Maturity models
- Fourth-generation languages
- Formal methods
- Agile methods
- Rapid and joint application development methods
- Rational Unified Process

Several models and methods have been developed that present techniques for solving RE problems e.g. Sommerville and Sawyer's (1997) Good Practice Guide and the unified approach of Leffingwell and Widrig (2000); Graham's Rapid Development (Graham 1998); Motorola's Quality Model Framework to strengthen the Requirements Bridge (Smith 1998) and methods for analysing and specifying requirements (Britton 1996; Britton 2000; Maciszek 2001; Kratochvíl 2002). These methods move away from 'what' constitutes RE best practice towards the more prescriptive world of 'how' to solve specific RE problems. The more prescribed the solution the less likely it is that the model is generic and universally applicable. Osterweil (1986) explains that one of the difficulties in relating the process to a problem is that all organisations are different,

"they differ in people, skills, products delivered, commercial and development strategies. Even within the same organization different projects present huge variations ... As a consequence, there is no unique, ready-made software development process. The process must be defined based on the problem to be solved" cited in Cugola and Ghezzi (1998, page 107).

This sentiment is echoed by Middleton and McCollum (2001) who point out that:

"The idea of 'best' method is misleading because of the diverse range of projects and developers. The generic lesson ... is that an organization is probably unwise to use a heavily prescriptive methodology to improve its software development performance" (Middleton and McCollum, page 18).

The lesson that can be learned from this RE research is that there is not a 'one size fits all' technique. The solution to the multifarious problems that organisations are having with their RE process is therefore likely to either be descriptive, generic and universally relevant, or prescriptive and detailed and relating to few organisations. What is lacking therefore, is a descriptive solution to the RE problem that encompasses both organisational and technical guidelines that can guide users towards the many prescriptive solutions.

2.5 The software process

This section presents an overview of the software process. It includes a brief background to Software Process Improvement (SPI), how SPI supports the RE process and some different approaches to software process improvement.

2.5.1 Taking a process view of software development

"Processes are like programs – you must get the specification and design right before you start worrying about optimisation" (Thomas and McGarry 1994, page 12).

Thomas and McGarry (1994) report that four out of five software development groups in their study had nothing they could describe as a process. Although this may not be representative of software development as a whole, it would appear that many software organisations are not aware of the importance of defining processes. Furthermore, this lack of process definition was also a finding in Hofmann and

Lehner (2001). Understanding and defining processes is perhaps the starting point for any quality improvement exercise. As Dutton (1993) notes

"process - the methods, tools, procedures, and techniques for developing and maintaining software - figures prominently in almost every discussion of software engineering" (Hofmann and Lehner 2001, page 56).

Cugula and Ghezzi (1998) explain that defining and controlling processes is important because they have a profound influence on the quality of products.

This emphasis on software processes is not new. The waterfall model, perhaps the first published software process model (Royce 1970; Royce 1987) appeared in 1970. In the following years, there have been numerous papers presenting alternative process models, such as the spiral model (Boehm 1988) Prototyping (Gomaa and Scott 1981) and the iterative enhancement model (Basili and Turner 1975). However, over the last decade the focus has shifted from the modeling of software processes towards the assessment of software processes and software management practices (Dutta and van Wassenhove 1997).

Deming, known for his pioneering work in quality management in post war Japan, declared that "if you can't describe what you are doing as a 'process', then you don't know what you are doing" (Deming 1900 – 1993). The Software Engineering Institute takes a similar view, describing software improvement as a journey where processes must be defined prior to implementing new methods, as "if you don't know where you are, a map won't help" (SEI 1996).

While it is possible that some organisations have decided not to take a process view of development, Humphrey argues that, "An orderly process framework is needed even when using the best programmers in the world" (Humphrey 1989). In a review of software processes Cugola and Ghezzi (1998) confirm Humphrey's view explaining that controlling processes has a profound influence on improving the quality of products.

It is likely that the idea of increasing productivity and quality through improved individual processes originated in manufacturing and the work of Shewhart (1931) in the 1930s. Shewhart's continuous view of process improvement was later adopted by Deming who applied his "Plan Do Check Act" cycle and statistical controls in both Japan and USA (Deming 1982; Deming 1986). This work, together with Juran's (1981) theory of quality controls and Crosby's (1979; 1986) five levels of total quality management, created the foundation for the process improvement model used in software development today. For example, Humphrey (1989) and his work with the Software Engineering Institute (SEI) applied similar five level process controls in his pioneering work on managing software.

2.5.2 Software process improvement (SPI)

"We must do more than create new techniques. We must understand the old ones" (Potts 1993, page 20).

In SPI the entire software task is treated as a process that can be controlled, measured and improved. Processes are defined as "that set of tasks that, when properly performed, produces the desired result" (Humphrey 2001). Many different SPI systems have evolved to support organisations in their improvement activities (Fox and Frakes 1997). These systems apply a cohesive set of theories, tools, methods and techniques in conjunction with attitudes, values and model problem solutions.

Baddoo (2002) describes SPI as having three primary goals;

- to improve the product through adopting practices within the development process such as 'requirements management' to reduce product faults, improve product maintainability, adaptability and usability and also satisfy user requirements.

- to improve process effectiveness, for example reduce timescales and shorten time-to-market, by taking a project management approach to development for increased control and transparency of the development process.

- to manage organisational change where key elements of change such as planning, implementation and communication are supported.

2.5.3 Approaches to SPI

Many process models have been developed to assist with SPI implementation. The following list shows the proliferation of software-related approaches to quality improvement as noted by Cottengim (2002), Thomson and Mayhew, (1997), Baddoo (2002) and Paulk (2001):

- ISO 9001
- ISO/IEC 15504 standard; SPICE (ISO 1999) and (SPICE 2003)
- Capability Maturity Model for Software (SW-CMM) (Paulk et al, 1995)
- Capability Maturity Model Integration (CMMI) (CMMI 2001)
- Joint Application Development (Wood and Silver 1995)
- Rapid Application Development (McConnel 1996)
- Quality Function Deployment (QFD 2003)
- Six-Sigma (Six Sigma 2003)
- The Malcolm Baldridge National Quality Award (MBNQA 2003)
- Ami handbook sponsored by EC ESPRIT programme (ami 1992)
- Bootstrap methodology (EC funded) see (Paulk 2001)
- TickIT (UK certification scheme) (Thomson and Mayhew 1997)
- Trillium (telecommunications product) (Thomson and Mayhew 1997)

Within the above list, the ISO 9001, ISO/IEC 15504 and the SW-CMM are some of the most applied approaches to SPI (Paulk 2001). The ISO 9001 is the most universally applied approach to SPI and as part of the ISO 9000 series of standards has been adopted by over 130 countries. One of the main limitations of the ISO series is that they are not industry specific (Ince 1994).

The ISO/IEC 15504 refers to a suite of standards on software process assessment developed by the International Organization for Standardisation (ISO) (SPICE 2003).

This work, often referred to as SPICE (Software Process Improvement and Capability dEtermination), focuses on software process issues but is also concerned with people, technology, management practices, customer support and quality. It harmonizes the many different approaches to software process assessment. The ISO/IEC 15504 is continuing to evolve as technical reports are trialled, and Paulk (2001) believes that the potential of this model is significant, noting that there are imminent plans for a release of an international standard.

History has shown that improvement models and management theories are discursive, "they come and go like waves on a beach and tend to reflect the dominant paradigm at the time" (Mullins 1993). This suggests that no single process improvement or management model provides all the answers – and if any of them do, it is all but transient. Current thinking, however, claims the Software Engineering Institute's (SEI) Software Capability Maturity Model (SW-CMM) as the de facto software process improvement standard initially in northern America and increasingly in developed countries throughout the world (Thomson and Mayhew 1997; El Emam et al. 2001; SEI 2003a).

The success of this SEI model is seen in the proliferation of CMMs developed by different groups for their different needs, e.g. (Burnstein et al. 1996). The future of the SW-CMM, however, resides within CMM-Integration (CMMI 2001), a new meta-model developed by the SEI to combine various CMMs. Goals of this CMMI meta model include

- Reconcile the architectural incompatibilities between CMM models
- Develop a meta-model that users can easily use to define CMM combinations
- Be capable of generating various versions of the CMMs.

(Source: Reifer 2000, page 97).

Paulk (2001) concludes that whichever approach is applied, to build competitive advantage, the focus should be on improvement and not on achieving a score, whether the score is maturity level, a certificate, or a process profile. This is easier said than done, as case studies show a marked difference in the scores achieved when organisations carry out confidential assessments compared with public evaluations (Baumert 1994).

The next section is dedicated to an in-depth study of the SW-CMM and CMMI models of SPI.

2.6 The Software Capability Maturity Model® (SW-CMM)

The SW-CMM (Paulk et al. 1995) follows an assessment methodology that aims to provide a roadmap to help organisations identify areas in the software process in need of improvement (Humphrey 1989). The model focuses on the capability of software organisations to produce high-quality products consistently and predictably. The SW-CMM presents sets of recommended practices in a number of key process areas (KPAs) that can enhance software-development and maintenance capability. Recommendations are based on knowledge acquired from software-process assessments and extensive feedback from both industry and the US government.

High level process maturity companies report the benefits of successful process improvement programs (Curtis 2000). Herbsleb and Goldenson's (1996) results show a correlation between higher maturity and meeting schedules, meeting budgets, improving quality, improving productivity; improving customer satisfaction; and improving staff morale. This is supported by (Humphrey et al. 1991; Herbsleb 1997; Rogoway 1998). Yet there is no guarantee that high levels of maturity necessary lead to better quality software (Pfleeger 1999). A recent study reveals that not all companies derive a competitive advantage when attempting to apply this improvement model (Rainer and Hall 2002). Moitra (1998) comments that this can be attributed to,

"the failure of organisations to clearly understand the crucial role of software processes in their operations ... ignoring the more important people processes" (Moitra 1998, pages 199-200).

Paulk et al (1995) acknowledge that although the SW-CMM directly addresses the human dimension only in training, people issues are not outside the scope of management responsibility or outside the scope of organisational needs. However, Paulk et al note that an effective process can empower people to work more effectively.

The framework was developed in the late 1980s at the SEI's Carnegie Mellon University in Pittsburgh, USA. Much of the initial investment and drive behind developing the framework came from the US Department of Defence (DoD) who needed a reliable method to help them select capable software contractors for their safety critical systems (Thomson and Mayhew 1997). The first description of this process maturity framework came in 1987 with the work of Humphrey and his team at the SEI (Humphrey 1988). The model continued to evolve over the next four years until in 1991 the SEI released its version 1 of the Capability Maturity Model for Software (SW-CMM) (Paulk et al. 1995). The model has now shifted from being used primarily by the DoD, as over 71% of companies reporting appraisals to the SEI represent commercial/in-house development. Also the offshore take up has increased, with over 119 appraisals in the UK and in excess of 250 in India also reported to the SEI (2003b).

The CMM describes an evolutionary path from an immature, ad-hoc software process to an optimizing, disciplined and mature process. The five stages of maturity and their associated key process areas (KPAs) are given in Table 1.

The software process maturity levels in Table 1 show 'the extent to which a specific process is explicitly defined, managed, measured, controlled and effective' (Paulk et al. 1995, p.9). In practice, the level of maturity is a measure of how successful a company has been in their software process improvement. Each Level (except Level 1) is deconstructed into several KPAs that indicate where an organisation should focus to improve its software process. KPAs identify the issues that must be addressed to achieve a maturity level. For example, if an organisation is at Level 3, it has addressed all Level 2 and 3 KPAs (with the possible exception of the Software Subcontract Management KPA that may not be applicable).

Table 1: SW-CMM 5 levels of maturity

Key Process Areas (KPAs)	Process characteristics	Action needed
CMM Level 1: Initial / Ad hoc Processes		
No recognisable KPAs at this level	Processes are chaotic and unpredictable. Few processes are defined, and success depends on individual effort and heroics.	Most important improvement needed is to institute basic project controls that require adequate preparation, clear responsibility, a public declaration and a dedication to performance.
CMM Level 2: Repeatable Processes		
Requirements management Software project planning Software project tracking & oversight Software subcontract management Software Quality assurance Software configuration management	Basic project management processes are established to track cost, schedule and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications. Processes are not consistent throughout the organisation.	Project management needs an understanding of the job's magnitude, senior management oversight and commitment, a quality assurance group to assure management that software work is done the way it is planned, and change controls.
CMM Level 3: Defined Processes		
Organisation process focus Organisation process definition Training program; Peer reviews Integrated software management Software product engineering Intergroup coordination	The software process for both management and engineering activities is documented, standardized and integrated into a set of standard software processes across the organisation.	Establish a process group, establish development process architecture, introduce a family of software engineering methods and technologies, e.g. design and code inspections, formal design methods.
CMM Level 4: Managed Processes		
Quantitative process management Software quality management	Detailed measures of the software process and product quality are collected. The organisation has a quantitative understanding and can control both the software process and products.	Establish minimum set of measurements to identify the quality and cost parameters of each process step. Establish process database and the resources to manage and maintain it. Provide sufficient process resources to gather and maintain this process data. Assess the relative quality of each product and inform management where quality targets are not being met.
CMM Level 5: Optimising Processes		
Defect prevention Technology change management Process change management	Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies	Process data is used to analyse and modify the process to prevent problems and improve efficiency. The data is available to justify the application of technology to various critical tasks.

(Source: Humphrey *et al.* 1989; Paulk *et al.* 1995; Paulk 2001)

2.6.1 SW-CMM-based appraisals

Methods for appraising the maturity level of an organisation in the SW-CMM come in two major classes: assessment performed for internal process improvement and evaluations performed by a customer. The two basic objectives of the SEI appraisal methods therefore are for self improvement and evaluation (Paulk 2001, pp 17-18):

a) The software self improvement process assessment is an appraisal by a trained team of software professionals to determine the state of an organisation's current software process, to determine the high-priority of software related issues facing an organisation and to obtain the organisational support for software process improvement.

b) The software capability evaluation (SCE) is an appraisal by a trained team of professionals to identify contractors who are qualified to perform the software work or to monitor the state of the software process used on an existing software effort. The SCE can be performed by an organisation, if they wish, for source selection and verification of another organisation's appraisal results/maturity level.

However, not all organisations conduct formal SEI appraisals and opt for self assessment of their processes. The SEI emphasise that the intended goal and purpose of their models and appraisal methods is for self improvement:

"The outcome, which is entirely dependent on the organization that follows these practices, is to raise the level of quality of the products developed with a better ability to predict the time and budget needed to develop the product. The goal focuses less on a perceived business advantage and more towards the ability to reliably develop products in a repeatable fashion with continual improvement versus doing the same in a chaotic state" (SEI 2003b, electronic source).

Another reason for conducting internal evaluations is that waiting 2 years between SEI formal assessments may be too long. For example Motorola designed their own process assessment that allowed them to monitor their process strengths internally at self-regulated intervals (Daskalantonakis 1994).

2.6.2 The SW-CMM and the CMMI

The SEI advocates the adoption of CMMI models and claims that they are "the best process improvement models available for product and service development and maintenance" (SEI 2003c). These models build on and extend the best practices of the SW-CMM, the Systems Engineering Capability Model (SECM), and the Integrated Product and Process Development Capability Maturity Model (IPPD-CMM) (SEI 2002b).

Maintaining a certain maturity status is a continuous process. Therefore once a certain level is reached, appraisals are still necessary to know if the maturity is being maintained over time. The SEI plan to phase out the SW-CMM as an independent model and integrate it with the CMMI which takes the following view of assessment:

"The Standard CMMI Appraisal Method for Process Improvement (SCAMPISM) is designed to provide benchmark quality ratings relative to Capability Maturity Model® Integration (CMMI®) models. It is applicable to a wide range of appraisal usage modes, including both internal process improvement and external capability determinations" (SEI 2003c, electronic source).

The assessment of maturity levels in the CMMI takes on two representations.

(1) Staged: similar to SW-CMM 5 level maturity framework shown in Table 1 - a complex methodology described in a 729 page report (SEI 2002a) ;
(2) Continuous: aligned to the process focus in the 15504 IEC/ISO. A complex methodology described in a 724 page report (SEI 2002b).

In the continuous representation in (2) above, practices are viewed in four groups according to their function: Process Management; Project Management, Engineering and Support (SEI 2002b).

The components of both the staged and continuous representations are similar comprising: process areas, specific goals, specific practices, generic goals, generic practices, typical work products, sub-practices, notes, discipline amplifications, generic practice elaborations, and references. As an example, the specific and generic practices for requirements management within both the continuous and the staged version are shown in Table 2. In the continuous version the practices come under the requirements management process, while in the staged version the practices span various maturity levels.

Table 2: CMMI Requirements Management Process Area

CMMI Goal	Practice	CMMI Requirements Management Practices
Goal 1		Requirements are managed and inconsistencies with project plans and work products are identified
	Specific	1.1 Develop an understanding with the requirements providers on the meaning of the requirements
	Specific	1,.2 Obtain commitment to the requirements from the project participants
	Specific	1.3 Manage changes to the requirements as they evolve during the project
	Specific	1.4 Maintain bi-directional traceability among the requirements and the project plans and work products
	Specific	1.5 Identify inconsistencies between the project plans and work products and the requirements
Goal 2		The process is institutionalised as a managed process
	Generic	2.1 Establish and maintain an organisational policy for planning and performing the requirements management process
	Generic	2.2 Establish and maintain the plan for performing the requirements management process
	Generic	2.3 Provide adequate resources for performing the process, developing the work products, and providing the services of the requirements management process
	Generic	2.4 Assign responsibility and authority for performing the process, developing the work products, and providing the services of the requirements management process
	Generic	2.5 Train the people performing or supporting the requirements management process as needed
	Generic	2.6 Place designated work products of the requirements management process under appropriate levels of configuration management
	Generic	2.7 Identify and involve the relevant stakeholders of the requirements management process as planned
	Generic	2.8 Monitor and control the requirements management process against the plan for performing the process and take appropriate corrective action
	Generic	2.9 Objectively evaluate adherence of the requirements management process against its process description standards and procedures, and address non-compliance
	Generic	2.10 Review the activities, status, and results of the requirements management process with higher-level management and resolve issues.

Source: (STSC 2003)

A review of how the CMMI maps to the SW-CMM reveals how many of the practices have been abstracted from KPAs in the SW-CMM. For example, many of the Requirements Management CMMI practices shown in Table 2 are also found in

Intergroup Co-ordination, Software Configuration Management and Software Product Engineering KPAs in the SW-CMM (STSC 2003). The CMMI also adds some processes not modelled in the SW-CMM. Although the CMMI builds on feedback from experts on where CMMs can be improved, Reifer questions whether this integrated approach includes too much information. It attempts to address all key practices required to help organisations improve their product and service development, acquisition, and maintenance processes. Reifer comments on the size of the document and considers it to be 'formidable' (Reifer 2000). However as it is still in the early stages of its release, it is possibly too early to assess how successful the CMMI is in its support of this wide range of inter-related software processes.

2.6.3 Strengths and weaknesses of the SW-CMM

While high level maturity companies report the benefits of successful process improvement programs using the SW-CMM (Curtis 2000), not all companies derive the benefits of this improvement model (Moitra 1998; Rainer and Hall 2003). For example Pfleeger alleges that

"the CMM is imperfect – there is no guarantee that a Level 5 organisation will produce good software. However, if we understand the uncertainty inherent in using the CMM, we can feel confident that a Level 5 organisation will produce good software a certain percentage of the time under certain conditions" (Pfleeger 1999, page 34).

Therefore, taking a circumspect view of the SW-CMM, I consider its strengths and weaknesses in order to assess how well this model is helping software companies to produce good software. Some characteristics of the SW-CMM are dichotomous and can therefore appear both a strength and weakness as detailed below.

2.6.3.1 Benefits of using the SW-CMM to support RE

The idea that the SW-CMM contains many RE related activities is encapsulated by Leffingwell and Widrig (2000), who believe that "The CMM moves the organization toward an integrated view wherein technical requirements must be kept consistent with project plans and activities"(page 458). Below are some of strengths noted both in the literature and by observation of SW CMM characteristics.

- The Requirements Management Key Practice Area (RM KPA)

The RM KPA addresses many of the specific problems identified in the review of RE. For example, technical support is shown in the specific activities that help steer practitioners away from vague requirements such as:

"The allocated requirements are reviewed to determine whether they are clearly and properly stated." (Paulk et al. 1995, pages 129-130)

Here, software engineering groups are directed to review the allocated requirements before they are incorporated into the software project. Incomplete and missing allocated requirements are thereby identified and the allocated requirements are reviewed to determine whether they are: feasible; clearly named; properly stated; consistent with each other; and testable. Management activities are also included in the RM KPA, where practitioners are guided to follow "a written organisational policy for managing the system requirements allocated to the software project" (Paulk et al, 1995).

- Process implementation prioritisation

The 'staged' structure of the SW-CMM guides requirements management by helping users to prioritise RE process implementation as recognised in the SW-CMM and the related IDEAL assessment model. McFeeley (1996) notes that

"The baselines, particularly the maturity baseline, typically identify issues and provide recommendations based on a much broader consensus than may have been available before. ... These issues and recommendations serve to provide some guidance, and often, a prioritization of actions" (McFeeley 1996, page 79).

- Goal Focus

Stating the goals of each improvement activity is integral to all KPAs. For example the RM KPA goals are explained as: Goal 1 "System requirements allocated to software are controlled to establish a baseline for software engineering and management use"; and Goal 2 "Software plans, products and activities are kept consistent with the system requirements allocated to software" (Paulk et al, 1995). Taking a process view of RE allows practitioners to work aggressively to achieve their goals (Paulk et al, 1995).

- Process abstraction

The SW-CMM advocates breaking down the software development into a limited set of activities. It does not prescribe a 'specific' path but guides users towards identifying and defining a software life cycle with predefined stages of manageable size (Paulk et al, 1997).

- Useability

The SW-CMM is the most applied software process improvement model, for example, (El Emam and Madhavji 1995b) state "The CMM has become a de facto standard as a basis for software process improvement". The strength of this wide application includes a growing familiarity amongst practitioners with the principles involved in this form of SPI that crosses organisational and departmental boundaries. Data on the number of appraisals reported to the SEI show a steady increase since its introduction in 1987 through to January 2003 (SEI 2003b). However, as many organisations conduct their own informal appraisals figures reported to the SEI are only a guide.

- SEI continues to support the CMM concept

The SW-CMM continues to be implemented and supported by the SEI is an added strength in the volatile area of software development. When a need is recognised an amendment or addition is made. The SW-CMM has been supplemented by other SEI improvement paradigms such as the IDEAL improvement model (McFeeley 1996), the People Capability Maturity Model (Curtis et al. 1995), Personal Software Process (Humphrey 1997); The Team Software Process (Humphrey 2000); and the Software Engineering CMM (SE-CMM 2003). This augmentation shows an on-going commitment to investment in the CMM concept.

- The SW CMM has a proven track record

It has been possible to track SW-CMM project records since its release in 1991 and empirically assess the benefits of this form of process improvement. Studies have shown that thousands of users have made significant improvements in product quality, productivity and cycle time through using the SW-CMM together with the CMM Based Assessment for Internal Process Improvement (Humphrey et al. 1991; Herbsleb 1997; Rogoway 1998; McConnell 2002). These studies indicate that increased process maturity as defined by the SW-CMM is indeed related to increased product quality. In their empirical study, El Emam and Madhavji (1996a) note that maturity measures are good predictors of organisational and project effectiveness. The SEI records that organisations' maturity profiles show a gradual shift towards companies achieving higher levels of maturity (SEI 2003b).

- The SW-CMM is tailorable

A proven strength of the SW-CMM is that it can be tailored to the specific needs of a company (Paulk et al. 1995) as shown in the many framework adaptations both inside and outside the field of software engineering. There are reportedly 34 CMMs developed by different groups using different architectures (Reifer 2000). Examples of model adaptation are included in the work of (Hackos 1997; Christie 1999; Potter and Sakry 2001; Ferraiolo 2002; Neissink et al. 2002). Indeed, the SW-CMM openly encourages development of new specialised, complementary models as using the SW-CMM's proven, and familiar framework will contribute towards user migration and understanding (Sheard 2001; Humphrey 2002).

- The SW-CMM is evolving

Despite SEI plans to cease developing SW-CMM as an independent model, the model continues to evolve (Conradi and Fuggetta 2002). The SW-CMM has become an integral part of the CMMI that addresses software, systems engineering and integrated product and process development issues. The CMMI attempts to bring the different improvement models together under one meta-architecture which users can employ to generate combinations of CMMs of interest to them (Reifer 2000) and (CMMI 2001).

- The SW-CMM is a recognised standard

SW-CMM maturity profiles help to build stronger customer-supplier relationships, for example Boeing has become increasingly reliant on the integrity of supplier software quality systems. Using the software process maturity level as a gauge, allows customers to make informed decisions in their choice of software supplier prior to making a commitment (Paulk, 2001).

2.6.3.2 How the SW-CMM fails to support RE

"All models are wrong; some models are useful."
G. Box cited in (Paulk et al. 1995) page 13.

A growing body of literature highlights some of the risks associated with basing improvements efforts on a model. This section notes some of the limitations that are due to how the SW-CMM is implemented as well as problems associated with the model's design.

- Complex presentation

The SW-CMM is a large and complex document that is difficult to understand (Paulk 2001). It is difficult to interpret as noted by Gilb, who expresses that models such as SW-CMM are "well-intentioned and contain some really good ideas. The problem is that they are often misunderstood, mistaught, and misapplied" (Gilb 1996). The SW-CMM includes many activities that, although related to software development is unhelpful when attempting to identify problems within the requirements process. Its complex nature may make it difficult for the non-expert to tailor or extend the SW-CMM (Paulk, 2001). Also, the number of activities and resources required appears to address the needs and budgets of large organisations performing contractual work (Paulk, 2001).

- Incomplete

Although the SW-CMM is complex some essential concepts are missing. Many of the activities necessary in a strong RE process are not all contained within the requirements management KPA (STSC 2003). As such the SW-CMM does not support the RE process (Sommerville and Sawyer 1997; Smith 1998). Researchers suggest that the SW CMM does not effectively deal with the social aspects of organisations as it lacks a managerial focus and should be supplemented with socially oriented theories (Ngwenyama and Nielsen 2003). Also, the SW-CMM maturity levels are gross measures of process capability and therefore oversimplify a complex set of issues (Baumert 1994).

- Prescriptive assumptions

The SW-CMM imposes a top down view of improvement where universal practices are presented that assume a connection between improved processes and improved product quality (Thomas and McGarry 1994). This calls into question whether products, goals, characteristics and local attributes of a software organisation are taken into account, without which it is not possible to guide the evolutionary process changes (Thomas and McGarry 1994).

- Weak links between software processes and improved performance

The SW-CMM is not a natural or essential representation of software processes (Bach 1994). El Emam and Birk (2000) note that the relationship between the SW-

CMM and performance remains a premise that enjoys weak empirical support for RE practices. Assessment depends on two assumptions:

a) That the practices defined in the assessment model are indeed good practices and their implementation will therefore result in improved performance.
b) That the quantitative assessment score is a true reflection of strengths and weaknesses in the process.

- Missing processes and broad assumptions

Process improvement goals and customer expectations are not adequately modelled; contradictory sets of assumptions about organisational culture; assumptions about the order of process implementation; and vague and incomplete sets of processes, e.g. (Brodman and Johnson 1994; Hayes and Zubrow 1995; Sommerville and Sawyer 1997; Lauesen and Vinter 2001; Hall et al. 2002a; Ngwenyama and Neilsen 2003).

- Emphasis on maturity recognition rather than improvement

The SW-CMM encourages displacement of goals from the true mission of improving the software process to the artificial mission of achieving a higher maturity level (Bach 1994) thus steering organisations towards process goals rather than meeting their business goals (Potter and Sakry 2001) and (Fayad 1997). The normative nature of SW-CMM based improvement can cause the organisation to neglect important non-CMM issues as noted in (Herbsleb 1997). The public evaluation of processes causes practitioners to be guarded in their assessments (Baumert 1994).

- An inflexible structure

The SW-CMM structure can cause the organisation to become rigid and bureaucratic making it more difficult to find creative solutions to technical and cultural problems as noted in (Herbsleb 1997; Ngwenyama and Neilsen 2003). Hather et al. (1996) note that a recognised weakness of the SW-CMM is that it does not take into account the ability of different processes to exist at different maturity levels. This may be due to the static nature of KPAs that do not provide an evolutionary view of processes, which would be of value to the individuals responsible for implementing, controlling, and improving a specific process (Paulk, 2001). Also Brodman and Johnson (1994) believe the SW-CMM favours the waterfall method and does not address prototyping. This structured view of development may run counter to agile methods that prefer not to be confined by predefined development stages. Also this presentation may be considered too rational and mechanistic to include an organisational culture perspective important in all models of change (Ngwenyama and Neilsen 2003).

- Appraisals

SW-CMM appraisals frequently do not result in action to address the problems identified. Appraisals often present a composite picture of process strength that does not reflect project level issues (Baumert 1994). Baumert continues, that making people responsible for process weakness is problematic as the natural tendency is to take credit for strengths and deny weaknesses. Also appraisals are often performed by untrained and unqualified appraisers, leading to inconsistent and unreliable appraisal results (Paulk, 2001). Finally, appraisals occur approximately every two years, which is considered too infrequent to highlight process problems (Baumert 1994; Daskalantonakis 1994). Yet, this does reflect the time it takes an organisation to move from one maturity level to the next.

However, Curtis (1994) provides some counter-arguments that address many of these criticisms:

- The SW-CMM is deliberately focused on the software process, and other factors can be addressed through other CMMs. Focussing on a vital few issues helps to identify improvement priorities that are generally true for any software organisation.

- The SW-CMM is structured hierarchically. The normative component is fairly short with 18 KPAs and 52 goals. The practices within each KPA are informative components that help CMM users interpret what is intended. The guidance in the key practices and sub-practices should be a help in understanding what a key practice or goal means.

- The SW-CMM explicitly describes organisational capability in terms of maturity levels.

- Training is available for assessors and evaluators from both the SEI and authorized distribution partners.

- The SW-CMM has been reviewed by thousands of software professionals as it has evolved.

There are, however, further reasons why the SW-CMM is failing the RE process. Nguyen and Swatman (2003) believe that the RE process differs from other software processes and therefore requires separate treatment. They call for a new process management approach to deal with the individual behaviour of the RE process that still needs to be monitored and controlled.

The need for a specialised model to monitor and control the RE process is recognised in the work of Sommerville and Sawyer (1997) with their RE Good Practice Guide. Sommerville and Sawyer's rationale for developing a three staged maturity model specific to RE is based on the assertion that the SW-CMM doesn't support RE. Their RE good practice guide includes an extensive catalogue of RE practices organized into a recommended order of implementation. Another good practice maturity model that includes RE processes is the emerging ISO/IEC 15504 international standard that addresses the RE problem in a defined "software requirements" process (El Emam and Birk 2000).

The weaknesses identified in the SW-CMM show how the RE process is not being supported. In particular aspects such as the SW-CMM's complexity and confused goals are inhibiting RE process improvement.

2.7 Principles involved in SPI modelling

The literature is examined to gain an understanding of the principles involved in SPI model development. Model development is an integral part of my research, and this knowledge is therefore used to guide the work. This section forms the concluding subject under review.

"Modeling is in the best tradition of science, because it helps us study phenomena closely" (Tichy 1998, page 32).

Simple models cannot precisely measure process maturity and complex models are not useful in guiding improvement. Yet simplicity enables engineers, managers, executives, and acquisition people to understand the framework, agree on where the organisation stands, and understand the needed improvements (Humphrey 2002). Creating separate models of complex activities in software development is likely to be more helpful than over-burdening one model.

2.7.1 Best practices in SPI

The importance of retaining a goal focus is a constant theme in the SPI literature. Cottengim (2002) suggests that organisations should relate their problems to goals asking questions such as "How does this problem fit into the organisation's larger mission? Can a link be drawn between solving a particular business need and a larger organisational performance goal?" McFeeley, with his work on the IDEAL model, warns that unless organisations are driven by current business needs that are understood and agreed to by management, it will be difficult to sustain the improvement program over the long haul (McFeeley 1996).

Process models are used in SPI to provide a more formal definition of the development of software and help to identify and validate metrics (Solingen and Berghout 1999). According to Madhavji (1991) SPI models should be customised to specific needs as defined by project goals. Model customisation is no simple task and requires characterising various aspects of the project (e.g. resource constraints); setting up project goals; assessing how these goals are supported by the adopted process model, tailoring the process model to suit project goals; using the tailored process model in the project; assessing and fine-tuning the model on an on-going basis.

"The customisation process would be simplified considerably if process models were organised hierarchically, leading from generic models at the top of the hierarchy to specific models at the bottom" (Madhavji 1991, page 237).

Models should reflect the natural order in process improvement. For example, Solingen and Berghout (1999) do not recommend basing improvement on a method that prescribes the installation of a software configuration management system, while most projects in the organisation fail because of bad RE management. Prioritising process implementation therefore requires recognising which processes need strengthening. A typical objective of a company engaged in a software improvement initiative is to document the current software process (i.e., 'as is' baseline) and define one or more ideal processes (i.e., 'to be' goal) to strive for (Krasner et al. 1992).

The model should be accessible to and understood by all key stakeholders, for example, project team members involved in RE, their manager and the improvement team members, who all need to be involved in the definition of measurement goals (Thayer and Dorfman 1990; Standish Group 1995; El Emam et al. 1996; Sommerville and Sawyer 1997; Boehm 2001; Hofmann and Lehner 2001).

SPI models should include process measurement as it "helps in making intelligent decisions and improving over time. But measurement must be focused, based upon goals and models" (Basili 1995b). Process measurement is dealt with in the next sub-section.

To summarise, SPI models should:

- be goal oriented
- be tailorable to meet needs of a specific project
- be understood by all key stakeholders

- have clearly defined processes that relate to key requirement engineering needs
- have a means of defining the current process within an organisation
- be simple to follow or navigate (not over-complex)
- have a clear means of assessing (or measuring) process strength
- have a well defined hierarchy that guides the user from a generic view of improvement practices through to prescriptive detailed guideline

Finally, rather than create a detached model to encapsulate these desirable qualities, Wiegers (1998b) suggests that model developers apply techniques defined by existing models and frameworks in a routine and effective way. Wiegers adds that only when the practical limits of known approaches have been reached, should we turn to improved models that provide guidance for working in better ways. Therefore, as the current SW-CMM approach to improvement seems to be 'necessary but not sufficient ...' and does not address many crucial processes in the area of RE (Rogoway 1998), there is a need to create a specialised SPI model to fill this gap.

(Humphrey 2002) also supports the re-use of solutions, stating:

"When faced with a problem software people generally find their own solutions, even when the problem has been solved many times before. The fact that it is so hard to build on other people's work is the single most important reason why software has made so little progress in the last 50 years" (Humphrey 2002, page 50).

Taking this advice, existing studies in the area of model development and adaptation will inform my research.

2.7.2 The Goal/Question/Metric (GQM) Paradigm

To be effective, a process improvement program must be accompanied by measurements to support them (Pfleeger 1995). Measurements and information fed back to developers, managers, customers and the corporation help in the understanding and control of software processes and products and the relationships between them (Basili 1995a). The GQM is a mechanism for supporting the setting of operational goals and is used for defining and interpreting software measurement within and across projects.

The GQM supports, complements and enhances SW-CMM process assessment. According to the creators of this paradigm, Basili and Rombach (1988), the GQM is aimed at providing a basis for corporate learning and improvement and has been used to guide software process improvement activities as shown in the work of (Pfleeger and Rombach 1994; Pfleeger 1995; De Panfilis et al. 1997; Mashiko and Basili 1997; Gresse and Briand 1998).

The GQM paradigm introduces goals prior to any data collection activities (Olsson and Runeson 2001). If improvement goals are not defined, an organisation's improvement activities will turn out to be as chaotic as the development process itself (Solingen and Berghout 1999). Although methods such as the SW-CMM stress the importance of characterising, classifying and decomposing goals they may have limited effect if there is no mechanism for reflecting on what appropriate goals to set in the first place (Antón 1996).

The GQM approach also relates measurement directly to the needs of the company. There is an ongoing debate as to whether companies should take a top-down or a bottom up approach to process improvement, e.g. (Thomas and McGarry 1994). Quality improvement models such as ISO9001, Bootstrap and the SW-CMM impose a top down view of improvement where universal practices are presented. Whereas according to Thomas and McGarry, organisations need to take a 'bottom up' approach as their goal should be 'product' improvement not 'process' improvement. The GQM takes a tailored approach to process assessment, as shown in the work of (Gresse and Briand 1998; Olsson and Runeson 2001).

Taking a GQM approach can help to identify a small area in software development in need of improvement, as shown in (Solingen and Berghout 2001). The GQM model appears flexible, allowing users to apply different levels of its methodology. For example, detailed use was made of the model in order to measure the requirements management KPA in the SW-CMM (Loconsole 2001), while on the other hand, Lavazza and Valetto (2000) used only the GQM 'plan' to help them measure the cost of requirements change.

Jalote (1997) notes that the GQM paradigm suggests a general framework for collecting data from projects that can be used for a specific purpose. The basic premise behind this approach is that there is no "general set" of metrics to be collected and an organisation must specify its goals before measuring anything. The GQM proposes that to start the measurement activity there must be a set quality or productivity goal at some level; this translates well to a set level of process capability.

Bache and Neil (1995) have identified several problems with the GQM approach, the two main issues being:

- Problems associated with top-down approach to problem solving: The GQM assumes that the problem to be solved is sufficiently well-defined to be decomposed into smaller units, which can be readily solved.

- Goal identification: The GQM assumes that goals have been correctly identified and that the metrics to support these goals can be readily defined and collected.

2.7.3 The model development process

This final section characterises some of the activities that underpin the modelling process. The importance of this work is highlighted by Eriksson, who states that,

"Since the quality of the model will affect the quality of creations that are guided by these models it is important to reflect upon the process of model construction" (Eriksson 2003, page 213).

According to Eriksson, there is very little in the literature to guide this work as "model development is an area in need of further research" (Eriksson 2003). Pidd also emphasises the need for this area to be given serious consideration since "learning the skills of modeling may be more important than learning about models", and pleas "for some serious research about how people go about their modeling" (Pidd 1999).

The word "model" can be defined as "a scaled down version of an object which reproduces a certain number of properties of the larger object on which it was based" (David 2000). The model should provoke thought and deliberation about the area in question. According to Pidd (1999), the modelling process involves capturing knowledge that is:

- external and explicit (allowing for external examination)
- a representation of the real world - a simplification of the world they represent
- a representation that is partial yet detailed enough to be useful and understandable
- governed by the intended use (e.g., to enable change, management and control of key RE processes at different levels of maturity)
- goal oriented

In their empirical study, Srinivasan and Te'eni (1995) considered the cognitive dimension to the model building process. The stages shown in their framework in Table 3 provides a generic and cyclical view of 5 processes involved when modelling:

Table 3: 5 cognitive stages of model building

Modelling stages	(Srinivasan and Te'eni 1995) modelling process
1	seek information about goals, objects and actions from external sources
2	translate the information from external sources in the light of previous knowledge
3	internally represent objects, relationships, actions and strategies
4	use tools to externally represent the objects, relationships and actions
5	test, and as a result, refine internal & external parts of problem representation

Although the work presented by Pidd (1999) and Srinivasan and Te'eni (1995) is useful as a general guide to the processes used and knowledge required in model development, there is no detail given as to how to initiate model development. According to (Rossi 1999; Koubarakis and Plaxousakis, 2002) model development is initiated by creating and agreeing model criteria. Objectives should be set to clarify the purpose of the model and to outline what the model is expected to describe. Having a clear set of objectives will help to steer model development and creates criteria against which the model can be tested for correctness and completeness (Madhavji, 1991). This formalises the model and sets out rules to create a firm foundation and provides a structure for the building process. This rule-based development technique is particularly relevant to the modelling of processes (Madhavji, 1991).

When populating the model, through an internal representation of objects, relationships and processes, Potts (1997) notes the tension between a model that is context specific and a model that is abstract and general. Potts states that if a model is based purely on abstraction it will have powerful properties such as the ability to generalise across contexts (i.e. the model will apply to more than one situation). However "Abstractionism provides standard methods, yet can also be an over-simplification of the problem domain with an overemphasis on normative cases". Potts adds that there are strengths to including context into the model as, "if the model is context specific it will fit in well with current practice and can be understood by end-users" (sic). But as Cugola and Ghezzi (1998) point out, moving away from abstract, normative models towards a context specific model involves following an expected sequence of activities. This limits flexibility and prohibits fast adaptation required in a dynamic marketplace.

However, Potts (1997) argues for retaining a context focus in modelling and addresses the field of RE practice where "by abstracting away from the context of an investigation, the designer too easily lapses into modeling only those things that are easy to model." Practitioners must be given the opportunity to take responsibility to ensure that "all requirements, particularly non-functional requirements, have been identified, are described correctly, and are fully detailed" (Middleton and McCollum 2001). Therefore when building a model of RE processes, all key processes must be included, even if they are difficult to define and measure.

A starting point for process improvement is to describe the current processes in software development. The process model inherent in this description is called a 'descriptive' model. "Describing a process means making the software process explicit. This involves modelling the actual software process using an appropriate process modelling methodology." Madhavji (1991) continues, "the central part of such a methodology that deals with the design of a process model needs to address the formalisms which may be used to represent process models. Several different formalisms have been proposed to address these needs." It appears from this early definition of modelling that Madhavji believes that descriptive modelling has some of the formal elements of the prescriptive modelling discussed in (Cugola and Ghezzi 1998).

2.8 Conclusion

The literature has been reviewed in order to gain a balanced understanding of the following:

2.8.1 The impact RE has on software development

The RE phase of software development continues to create problems for software organisations. Indeed, it appears to be the main problem in software development. The traditional phased view of RE is not necessarily helping developers with their agile development methods. Yet, controls and measures are needed to manage the many processes associated with this area of development. The complex needs of the customer must be understood and engineered in any of the chosen development methodologies.

A key issue suggested by the literature is that the lack of support for 'organisational' processes is hindering practitioners in their RE activities. If RE is not given the structure, responsibility, resources, training and skills it needs all the technology and tools available are not going to help. RE needs a reliable way to ensure that both organisational and technical processes are in place and are at a required standard.

2.8.2 RE process problems and solutions

The literature indicates that the RE process requires a structure that can be controlled, although there is a debate about the universal applicability of the lifecycle models. Viewing RE as a series of inter-related processes that organisations can apply in their software development appears a workable option, as presented in CMMI RE process activities in Table 2.

There is no consensus as to the key RE problems in the literature despite many of the studies basing their findings on empirical research. This may be due to the differences in samples (e.g. single case studies, European-wide surveys; company size and function) and forms of analysis. However, each finding makes a contribution towards a broad understanding of the problems practitioners are experiencing in RE.

The literature indicates that although organisations may have common problems, they are likely to have different priorities. It is especially important that RE priorities and policies are considered as it is likely that the solutions offered may conflict (Lindland et al. 1994). Making improvements to the RE process will therefore require organisations to have an understanding of their own RE problems and improvement needs before seeking solutions.

2.8.3 Methods to assess software process strengths and weaknesses

The advantages of viewing RE in terms of processes is well documented and is based on the premise that an improved process will result in a higher quality product. One of the benefits of taking this approach to development is that processes can be measured, controlled and improved. The proliferation of software process improvement models reflects the lack of a standard that suits the many different development methodologies.

An analysis of the SW-CMM has been undertaken as an example of how this popular model of SPI approaches process assessment. The SW-CMM does not easily allow companies to isolate the RE process from the software process to establish where strengths and weaknesses lie. Although the SW-CMM has a KPA that focuses on requirements management, there are other activities that directly affect the production of requirements that have not been included in this area of improvement. For example the new CMMI in its representation of the Requirements Management process includes activities from the SW-CMM configuration management, software process engineering and KPAs. Also, companies must wait approximately two years between formal appraisals.

The literature suggests there are several ways to address the current weaknesses in the SW-CMM appraisal methods. These methods include creating an internal assessment model to supplement the SEI formal appraisal and adapting the GQM.

2.8.4 SW-CMM support for the RE process

The strengths and weaknesses of the SW-CMM detailed in the literature reflect the many advocates and opponents of its methodology. While there are undoubted weaknesses attached to the model design, the framework, which has evolved from the work of quality and management experts in manufacturing, has proven strengths. The many adaptations indicate that it is possible to apply this framework to focus on different development areas. Also, if a specialised software model follows the same structure as the SW-CMM, users of the model will benefit from a view of development that is integrated with the software process.

2.8.5 Developing a best practice model to support SPI

The principles behind model development are explored in order to inform this study. The main areas to consider when building a specialised process maturity model are:

- To maintain a balance between a complex model that can precisely measure the process maturity and a simple model that can easily be used to guide improvement. To follow a rule-based scheme that will guide development and create criteria against which the model can be validated. The model must have sufficient detail to allow it to be tailored to meet specific project needs and general enough to apply to a large section of the software community. The model must be simple enough to be interpreted by all key stakeholders who should be able to interpret the representation of processes with minimal training.

- To retain a goal focus that is based on a hierarchical structure to reflect the natural order in process improvement. The structure should be consistent with existing frameworks, as it is better to build on proven techniques rather than expect users to learn new techniques and concepts. This includes the process measurement or assessment methods that should also be tailored to meet the specific needs of the organisation.

2.8.6 Building on the literature

The studies performed in this thesis contribute towards a further understanding of RE process problems as detailed in the literature. I aim to complement the existing knowledge by exploring some of the issues raised that particularly focus on RE problems and solutions. The literature on the SW-CMM guides my model development where strengths and weaknesses are considered. I use the literature to define my work so that it can be integrated with the RE and SPI body of knowledge.

It is in the next section that I examine methods for collecting and analysing empirical data that will provide a fuller picture of the RE process problems and how they might be addressed in a specialised software process improvement model.

Chapter Three: Methodology

"Real-world problems are seldom where you expect them to be. A careful analysis of the problems that practitioners face often reveals that what the researcher thinks is a major practical problem has little significance; whereas, the neglected problem often turns out to be important" (Potts 1993) page 20.

Although I cannot claim RE to be a neglected area of research, it is clear from the literature that further work is required to pinpoint where support is needed from a practitioner's perspective. As Zelkowitz *et al.* (1998) point out, practitioners and researchers often have different ideas about what constitutes good evidence. This chapter presents the methods used in this research as detailed in chapters four, five, and eight. The methods encompass both the practitioner's preference for qualitative studies performed in context, and the researcher's preference for quantitative forms of analysis (Pfleeger 1999).

This methodology chapter explains how data is collected and analysed in this study and is structured as follows:

Section 3.1 introduces the concept of empirical research and places it in context with software engineering. This introduction to the methodology highlights some of the advantages and disadvantages of using techniques more usually associated with the social sciences. Section 3.2 explains how empirical research can be viewed as taking either a qualitative or a quantitative approach, and how both these approaches apply to my work. These two different approaches are shown to be complementary. Section 3.3 gives a rationale for the choice of my research design and focusses on data collection methods that include focus groups and questionnaires used in the three studies contained in this thesis. This section places the sample in context with the population to give an indication of how representative the data is of the population as a whole. The emphasis is on providing a context for the data. Section 3.4 explains some principles and objectives of conducting a pilot study. Section 3.5 gives a rationale for using an expert panel in a validation exercise. Sections 3.6 and 3.7 discuss data analysis methods that build on the data collection methods. These analyses include qualitative methods such as grounded theory and content analysis, and quantitative methods such as contingency tables and correspondence analysis. The questionnaire analysis methods are included in section 3.7, showing how results are presented within confidence limits. Finally, in section 3.8, I summarise the methods used in this study.

3.1 Empirical research and software engineering

"Science and engineering research fields can be characterized in terms of the kinds of questions they find worth investigating, the research methods they adopt, and the criteria by which they evaluate their results" (Shaw 2002) page 1.

I take a scientific approach to the analysis of the RE process where I collect and analyse empirical data in order to create and test a number of research questions. I try to adhere to the guidelines offered to software researchers in Kitchenham *et al.* (2002b) where the authors call for empirical methods to be reported in sufficient detail to allow for assessment, and for statistics to be used appropriately. Strok (2003), reporting from a symposium held during the recent international conference on software engineering, notes that it is the 'approach' to research that is important, "namely, how to formulate and validate hypotheses, most often through experimentation" (Strok 2003) page 93.

Empirical research, including both qualitative and quantitative methods presented in this study, have been widely used over many decades in the social sciences, e.g. (Maxwell 1975; Marshall and Rossman 1989; Bryman 1996; Ott *et al.* 1999; Denzin and Lincoln 2000). These research methods constitute a large body of well-defined, mature empirical methods that, as yet, have not been fully employed in the study of software engineering (Tichy *et al.* 1995; Walker *et al.* 2003).

As a starting point to this section, I consider methods suited to exploring how to meet the aim of the study, which is to create a RE process model that captures the needs of practitioners. The empirical research approach is used to gain a better understanding of the RE process. According to Black (1999), the term empirical research indicates that *"information, knowledge and understanding are gathered through experience and direct data collection"* (Black 1999, page 3).

The empirical research in this study makes direct observations that reflect practitioner experiences with methods, tools and techniques in a way that relates more to the real world than other research approaches (Harrison *et al.* 1999). For a socio-technical discipline like RE, empirical methods are crucial. This is because empirical methods allow the researcher to incorporate multidisciplinary and interdisciplinary factors that frequently arise such as human issues, communication difficulties, quality of processes and products (Wohlin 2003). Wells and Harrison (2000) concur, noting that it is becoming increasingly important to use empirical methods to further our understanding of human issues in software engineering.

Latterly, Fenton (2001) reports that empirical methods are also needed to create a more rational basis for decision-making and is particularly concerned about the lack of any empirical basis for decisions affecting all aspects of the software life-cycle. In 1986 Conte *et al.* asked that an empirical body of knowledge be built that is based on relevant quantitative information about real projects. It appears that earlier calls for the software engineering community to take a more scientific approach to their research (Fenton *et al.* 1994) and evaluate their ideas in a practical setting before advocating them (Glass 1996) remain unsatisfied.

3.1.1 Limitations of empirical research

Empirical investigation can help to uncover disparities between widely held assumptions and objective data, but it is not a panacea (Tichy 1998). Potts (1993) places the power of empirical research in context, stating that empirical research is not effective without a follow up as results of an empirical investigation are not ends in themselves. Software engineering is an action-oriented discipline; mere study is no substitute for improvement. Results may suggest a change in direction in technology construction or methodological practice, for example. Potts also warns that empirical research is not always reliable.

Potter argues that 'empirical' is the most problematic term used in research (Potter 1996). Potter argues that the all encompassing dictionary definition that "the practice of emphasizing experience, especially of the senses or the practice or method of relying on observation, experimentation, or induction rather than upon intuition, speculation, dialectic or other rationalistic means in the pursuit of knowledge" leaves him wondering what 'non-empirical research' is.

The term 'empirical research' is used in this study to represent a methodology based on direct observation; this is clearly different from theoretical research where inferences or deductions are likely to be based on anecdote or secondary sources. The empirical research methods in this study encompass both qualitative and quantitative methods. The next section gives a rationale for the use of these methods.

3.2 Qualitative and quantitative methods

Despite Einstein's (1879–1955) early recognition that, ''Not everything that can be counted counts, and not everything that counts can be counted,'' many researchers believed that the only phenomena that mattered were those that could be measured (Tesch 1990). In the past, quantitative researchers who used measurement to test their theories, took a condescending view of qualitative research that relied on observation, listening and interpretation (Miller *et al.* 2002). Webb (1999) takes up this theme, noting the difficulty in persuading granting bodies of the acceptability and rigour of qualitative research in comparison with quantitative approaches and strategies.

This difference of opinion appears to be narrowing as practitioners of empirical methods begin to realise that both approaches offer valuable insights. While quantitative analysis can answer many types of questions such as *when* and *how who* did *what* and *where*, it tends to ignore the more qualitative question of *why*. As a result my research is designed to include both methods in order to provide a context to empirical quantitative findings (Miller *et al.* 2002).

3.2.1 Qualitative Research

The complexity of the qualitative research field is illustrated in (Potter 1996; Shaw 1999; Denzin and Lincoln 2000) where authors appear reluctant to give a definition of the term. It is clear that qualitative research means different things to different people (Potter 1996). Tesch (1990) alleges that "strictly speaking, there is no such thing as qualitative research. There are only qualitative data". However, it is widely agreed that the term 'qualitative research' represents a certain approach to knowledge production, and 'qualitative data' is understood to mean any information the researcher gathers that is not naturally expressed in numbers (Tesch 1990; Seaman 1999).

Within the context of this study, however, qualitative research is understood to be:

"A 'holistic' overview of the culture and context under study .... where the researcher attempts to capture data on the perceptions of local actors 'from the inside' ... where a quality approach can effectively give voice to the normally silenced and can poignantly illuminate what is typically masked" (abstracted from several definitions given in Shaw 1999, page 13).

Qualitative research is conducted by researchers who want to examine some phenomenon, develop insights, and report those insights to others (Potter 1996). It is concerned with discovering causes as noticed by the subjects in the study, and understanding their view of the problem (Greenhalgh and Taylor 1997). Context therefore becomes central to the research as qualitative research is concerned with studying objects in their natural setting (Wohlin 2003).

Shaw (1999) states that qualitative evaluation is interpretative. Qualitative research begins with the acceptance that the phenomena can be interpreted in a variety of ways. It involves the studied use and collection of a variety of empirical materials such as case study; personal experience; interview and survey. It deploys a wide range of interconnected interpretive practices, hoping always to get a better understanding of the subject matter at hand (Denzin and Lincoln 2000). It is therefore important to consider which method to use as each practice could make the world visible in a different way.

A drawback is that qualitative analysis methods are generally more labour-intensive than quantitative methods. Qualitative results are often considered 'softer' or 'fuzzier' than quantitative results, especially in the technical software engineering community. Results are therefore more difficult to summarize or simplify (Seaman 1999). However, the results have a strong external validity that is often lacking in more statistically rigorous quantitative methods (Briand in (Walker *et al.* 2003)).

3.2.2 Quantitative research

According to Wohlin (2003) quantitative research is mainly concerned with quantifying a relationship or comparing two or more groups where the aim is to identify a cause-effect relationship. Quantitative data are typically represented as numbers, e.g. counts or measurements and therefore promote comparisons and statistical analysis (Seaman 1999). Quantitative research is often conducted through setting up controlled experiments, collecting data through case studies or surveys. It is therefore the data that is collected that can be described as quantitative or qualitative research as the methods of both overlap, e.g. surveys.

The use of quantitative research methods is dependent on the application of measurement (Wohlin 2003). The quantitative approach follows scientific traditions of induction and deduction and is often oriented towards searching for aggregate patterns across empirical observations (Potter 1996).

3.2.3 Objectivity in research

According to Shipman (1997, page 18) "It is easy to detect subjectivity in social research. It is impossible to confirm objectivity". Such a sentiment is echoed by Webb (1999) who notes that researchers are always steeped in expectations. Despite this shortcoming, the researcher must aim to be objective and detached. Recognising whether data is subjective or not is totally unrelated to the type of data collected. In other words, qualitative and quantitative data is connected to how the information is represented, not whether it is subjective or objective. Qualitative data is often assumed to be subjective and quantitative data is often assumed to be objective. Neither is necessarily true. In fact, "the objectivity or subjectivity of data is completely orthogonal to whether it is qualitative or quantitative" (Seaman 1999, page 563).

One method of determining the objectivity of research is based on peer review, where the researcher undergoes "friendly –hostile" assessment (Shipman 1997). But a major difficulty with this process is that, "different research communities will come to different conclusions. ... Who are the peers? What if they are the established, or a clique who know the author?" (Shipman 1997). A further method of regulating the level of subjectivity is through replication of the study as this is deemed to be "a means of checking the biases of the investigator" (Bryman 1996).

3.2.4 Combining research methods

Quantitative and qualitative methods are complementary (Briand in (Walker *et al.* 2003)). Drehmer and Dekleva (2001) concur, noting that quantitative science often begins with identifying conditions which, when observed, are deemed worth counting. Therefore qualitative data can be converted through coding to become frequency data, and hence quantitative. Seaman (1999) adds that although this process of coding transforms qualitative data into quantitative data, it does not affect its subjectivity or objectivity.

Bryman (1996) notes that although the combination of methods is more usually associated with a preliminary qualitative investigation, the reverse can also occur. The literature shows that initial examination of quantitative data can lead to an in depth qualitative study. For example, "one of the ways in which quantitative research can facilitate qualitative research is by the judicious selection of case studies for further research" (Bryman 1996) page 136.

Finally, this overview of methods indicates that empirical methods are particularly relevant to analysing the multifarious SPI problems software practitioners are experiencing in their daily activities. In line with recommendations, this research uses a combination of empirical data collected from software practitioners and experts in the field of SPI and RE. The description of the research design that follows, involves the collection of qualitative data that is transformed to quantitative data (i.e. frequencies) in order to carry out statistical analyses. Quantitative data is also collected in order to validate the results of the qualitative study. However, throughout the reporting of the results an effort is made to maintain the integrity of the original data.

3.3 Data collection methods

This study includes a mixture of research approaches where qualitative methods are used to define conditions and preliminary questions which can later be addressed in quantitative studies (Greenhalgh and Taylor 1997). As qualitative data are richer than quantitative data, using qualitative methods increases the amount of information contained in the data collected. It also increases the diversity of the data and thus increases confidence in the results (Seaman 1999).

The three separate studies in this research collect data from a variety of sources in different settings and apply a range of methods for both data collection and analysis. This section starts with an explanation and a justification of the data collection methods used, as these significantly influence the data analysis process that follows.

This research was initiated through two forms of data collection:

- Data is collected through a literature review that provides a context for the study by creating a synthesis of existing knowledge and solutions (Hakim 1987). This is covered in the preceding chapter that focussed on the RE process, SPI and modelling literature.

  This is in line with Kitchenham et al's (2002b) guideline that the relationship between the current research activity and other research should be defined, "so that researchers can combine to build an integrated body of knowledge about software engineering phenomena".

- Empirical data is collected from Focus Group transcripts.

  The studies in this thesis are grounded in the information elicited from focus groups.

An analysis of both the literature and the focus group data underpin model development. Further data collection was then performed in order to validate the model:

- Empirical data is collected from an Expert Panel Questionnaire
- Pilot Studies

3.3.1 Research context

In research it is important to define contextual information to allow researchers to compare and contrast the work with other studies. However there are currently no set standards in software engineering for determining what should be included in this definition (Kitchenham *et al.* 2002b). As such, this research observes the guidelines given by Kitchenham *et al.* (2002b).

To place this study in context, I include with each data collection method the following:

1. The population from which the subjects and objects are drawn
2. The process by which the subjects and objects were selected
3. The process by which subjects are assigned to treatments
4. The limitations of the process. These are not necessary exhaustive, but all those known are included.

3.3.2 Focus groups

Focus groups provide an empirical method for collecting qualitative data on how people in a particular setting come to understand, account for, take action and otherwise manage their day to day situations. It is therefore an ideal method to explore RE process problems. According to Shaw (1999), "One of the more promising developments in applying qualitative and participatory evaluation has come through the work on focus groups". Focus groups are a well documented social science research technique (Morgan 1997). They involve assembling small groups of peers to discuss particular topics. Discussion is largely free-flowing, but is directed by a researcher. Focus groups particularly allow human issues to be explored and have been described as "a way to better understand how people feel and think about an issue .." (Morgan 1997).

According to Shaw (1999), focus groups have three particular advantages:

a) The group interaction is itself the data, where the method enables the researcher to examine people's different perspectives as they operate within a social network and to explore how accounts are constructed, expressed, censored, opposed and changed through social interaction.
b) Focus groups are a form of participatory evaluation. Dividing groups into their specific role, or power group has particular advantages in highlighting differentials between participants and decision-makers. Implementing focus groups in this way has considerable potential for application in software engineering where groups are naturally divided into practitioner roles.
c) They introduce a valuable approach to learning the extent of consensus on a particular issue, where "the co-participants act as co-researchers taking the research into new and unexpected directions and engaging with each other in ways which are both complementary ... and argumentative" ((Kitzinger 1994), cited in Shaw 1999, page 156).

The case for using focus groups is summarized by Gibbs (1997):

"Focus groups are particularly useful when there are power differences between the participants and decision-makers or professionals, when the everyday use of language and culture of particular groups is of interest, and when one wants to explore the degree of consensus on a given topic" ((Morgan and Krueger 1993) cited in Gibbs 1997 electronic source).

Shaw (1999) recommends that separate sessions are held with homogeneous but contrasting focus groups. This division is believed to produce information in greater depth than would be the case with heterogeneous groups.

A further recommendation is made by Greenbaum (1998) who states that using external researchers to conduct focus groups is likely to result in a more objective data collection than using internal moderators as there is less investment in the groups' outcome.

3.3.2.1 Limitations of focus groups

Focus groups rely on interaction within the group based on topics that are supplied by the researcher. There is little or no research evidence on the relative benefits of focus groups against interview methods (Shaw 1999). According to Shaw (1999) there are some situations where focus groups should not be used:

- If the intention is to improve practitioner's participation or group skills
- For therapeutic purposes
- To secure immediate action
- If information, understanding or explanation is not central to the study

There are also practical constraints to be considered:

- Can personal views be readily expressed?
- Are breaches of confidentiality likely to be a problem?
- Is the group mixed in terms of authority or roles? (e.g. placing patients with carers, or senior managers with developers).

When planning focus groups these limitations should be considered as they are likely to compromise the data generated.

3.3.2.2 Implementing focus groups in this study

The RE process is very dependent on soft issues and focus groups are, therefore, ideal for exploring the problems companies experience with the RE process. Focus groups are particularly appropriate for the study of RE process problems as they elicit data which allows a better understanding of the differences between groups of people (Morgan 1997) and, more specifically, they can help to explore the different RE experiences and opinions of developers, project managers and senior managers.

According to Greenbaum (1998) when interpreting focus group data the researcher should not focus on individual participants, but should use the data to gain an impression of what 'the group' feels about an issue. Ideas should be analysed with the knowledge that one person's comment may not represent a consensus. For this reason the qualitative data analysis in this thesis is based on categorising problems and noting their frequency prior to interpreting the results.

3.3.3 Questionnaires

Questionnaires provide a further empirical method for collecting quantitative or qualitative data to evaluate RE process problems. They are a popular device for the measurement of concepts (Bryman 1996) and are multi-purpose in that the design can be adapted to almost all research topics (Hakim 1987). Hetzel (1995) in his chapter on 'the sorry state of software practice measurement', notes that questionnaires, or surveys, offer one of the most flexible means of analysing and better understanding process issues and practices (page 100). A strength of the questionnaire survey is that all respondents receive the same set of questions overcoming problems of replication present in more qualitative forms of data collection. A further benefit of a questionnaire survey is its transparency or accountability where methods and procedures used can be made visible and accessible to all parties (Hakim 1987). Mailed questionnaires also allow subjects to participate over a large geographical area.

The detached quality of the questionnaire leads to less biased data as the respondent is not influenced by the attitude or opinion of the interviewer or *vice versa* (Baddoo 2002). Using the questionnaire as my survey instrument also allows me to pre-determine the time commitment required of the respondents to complete the questionnaire (through a pilot study). However, with mailed questionnaires, although the initial costs of paper and postage may be low, the costs of follow-up and non-response may be high (Fink 1995).

There are other potential weaknesses in this form of survey. For example, it is possible that the respondents interpret the questions incorrectly and the questions may not capture the real issues under investigation. Also, when a questionnaire is structured and quantitative, the respondents are restricted in the level of detail they can supply, meaning that there may be "some loss of sensitivity and quality as compared with depth interviews" (Hakim 1987). However, this degree of loss depends very much on how a questionnaire is designed and carried out.

Results from my questionnaire are used to indicate possible strengths and weaknesses within the RE process model. Results are also used to generate theory where I consider the wider implications of experts' attitude to the SW-CMM and the RE process.

3.3.3.1 Questionnaire design in this study

The questionnaire provides an appropriate methodology to validate how well the RE process model reflects the needs of software practitioners. Alternative qualitative methods such as direct observation, experiment, semi-structured interview or case study are not as appropriate for this embryonic stage of development (Rodeghier 1996). At this exploratory stage I need to replicate questions directly associated with my model criteria. I therefore chose the questionnaire as my primary data collection method as it is best suited to the nature and type of data that I need to generate and analyse.

Using a mailed questionnaire also has practical advantages as I was able to invite experts to participate from dispersed geographical locations. As I invited the participants to take part, my sampling method could be regarded as a convenience sample, although the sample was drawn from experts who I was unsure would have the time or interest to participate.

3.3.3.2 Questionnaire response categories

Kitchenham *et al.* (2002b) emphasise the importance of relating outcome measures to the objectives of the study. In a similar study to my own, Dybå (2000) considers the relative merits of different measurement scales and concludes that a 5 point attitude scale is the most reliable measure, whereas El Emam and Birk (2000) use a 4 point attitude scale in their validation questionnaire. I use both scales in the questionnaire to suit the granularity of response required as in the examples given in Figure 1 (using a 5 point scale) and Figure 2 (using 4 point scales). When using a bi-polar form of questioning I employ a 5-point scale as shown in Figure 1.

	Too few		Correct number ← →		Too many	No opinion
	(1)	(2)	(3)	(4)	(5)	
"Does the model have the right number of processes at this level?"	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Figure 1: Example of a bi-polar 5-point attitude scale

For the majority of the questions I use a 4 point scale, as this even number allows me to dichotomize the responses. I plan to interpret the responses as either supportive or critical of the model as shown in Figure 2. Although this conversion results in a slight loss of information, viewing responses in two categories eases interpretation and analysis of the data. Collapsing the responses in this way has the advantage of taking some of the 'subjectivity' away from the analysis as, arguably, one person's 'strongly agree' may be another person's 'agree'.

QUESTION TYPES	CRITICAL RESPONSES		SUPPORTIVE RESPONSES		NEUTRAL RESPONSE
"The guidelines given are relevant to requirements engineering activities"	Strongly disagree (1)	Disagree (2)	Agree (3)	Strongly agree (4)	No opinion Missing Don't know
"How consistent is the level of detail given within the Requirements CMM?"	Not at all (1)	(2)	(3)	Very (4)	No opinion Missing Don't' know

Figure 2: Example of a dichotomised 4 point attitude scale

Figure 2 shows the supportive responses as categories (3) and (4). Tables in the results section of this thesis generally show supportive responses as either {'Agree'; 'Strongly Agree'} or {'(3)'; 'Very'}. However, to avoid the problem of 'participant acquiescence' I sometimes reverse the supportive response categories (Oppenheim 2001) i.e., some questions are designed so that a response of {'Disagree'; ' Strongly disagree'} and {'(2)'; 'Not at all'} are supportive.

3.4 Pilot study

According to Oppenheim (2001), in principle almost anything about a questionnaire can and should be piloted, to include type and colour of paper used and font size. It is essential to pilot every question, every question sequence and every scale in the study. Nothing should be taken for granted. The question of layout and even the question numbering system should be piloted. In the case of open questions it is not only important that the question is understood but that the coding and quantifying of the responses are explored.

Prior to releasing their questionnaire, Berry and Jeffery (2000) ran a test on each item in order to assess respondents' level of understanding, level of knowledge, level of difficulty in responding and level of relevance to subject area. I dealt with these 4 points through examining the pilot test responses and making changes as a result of the feedback.

3.5 Expert panel

- The population from which the subjects are drawn

According to Hakim (1987), small samples can be used to develop and test explanations, particularly in the early stages of the work. Previous studies have used small samples to gain expert feedback to evaluate and support model development. For example, Dybå (2000) used 11 experts to conduct his review process, and El Emam and Madhavji (1996b) interviewed 30 experts to elicit criteria for their instrument to evaluate RE success. The value of expert knowledge is also recognised in a recent evaluation of software quality that suggests methods to formally capture expert judgement (Rosqvist *et al.* 2003).

The reliability of using expert judgement is shown in other work. For example, Lauesen and Vinter (2001) found that the ability of experts to predict techniques to prevent requirements defects were very high when put into practice. Another positive outcome is observed in the work of Kitchenham et al (2002a) in their analysis of the accuracy of several methods of estimating project effort. Their statistical analysis revealed that a human centred estimating process incorporating expert opinion can substantially outperform simple function point models.

- The process by which the subjects were selected

Model validation is defined as "the process of ensuring that the model is sufficiently accurate for the purpose at hand" (Robinson 1997). My validation of the model aims to provide answers as to whether the right model is being built (Boehm 1981). At this stage of development I am not looking to verify whether the model directly meets the needs of its users, where I test whether this largely generic model meets the initial criteria for building the model in the first place. I do not therefore set out to directly evaluate the model's quality, usability and utility (Gass 1983) as at this stage of development I cannot test whether the conceptual model has been transformed with sufficient accuracy (Robinson 1997).

I emulated previous studies that validated improvement models and measurement 'instruments' by inviting a panel of experts to complete a detailed questionnaire, see for example, (El Emam and Madhavji 1996b; Dybå 2000; El Emam and Birk 2000). I targeted experts from different backgrounds and audience groups as recommended by Lauesen and Vinter (2001) and Kitchenham *et al* (2002). Experts were drawn from a population of experienced practitioners and researchers in the areas of CMM software process improvement and RE. I directly targeted this group to ensure that my sample has representatives from four areas of expertise as given in Table 4.

	SW-CMM	Requirements Engineering
Practitioners	✓	✓
Researchers	✓	✓

Table 4: Four areas of expertise represented by the R-CMM validation panel

These areas of expertise are represented to ensure that in the early development of the model, practitioner needs and researcher knowledge are fed back to the development cycle. SEI recommendations for experts participating in a process assessment support this cross-section of knowledge as shown in Figure 3.

SEI suggest the following team take part in process maturity assessment	Creating a team to validate the R-CMM comprises the following:
EXPERIENCED PEOPLE	
knowledgeable in the process	knowledgeable in the CMM method and/or requirements process (researcher)
knowledge in the technology (software development: coding, design etc....)	knowledgeable/practiced in requirements (elicitation, specification, validation: traceablily, modelling etc.) (practitioner)
knowledge of the application area	Participated in SPI: process assessment; modelling; measuring Participated in requirements activities (practitioner)

Figure 3: Adapting SEI assessment team recommendations to requirement validation team attributes (Paulk *et al.* 1995)

3.5.1 Sampling considerations

There are several factors that affect the amount of error or chance of variation in the sample. Factors that most influence sampling variability are the sampling method, the sample size and the response rate.

- The quality of the survey's findings may be decreased if the sample design deviates from a random sample; or probability sampling, relying on convenience sampling. Ideally samples will be chosen through a random sample method although, other methods, such as stratified random sampling, can be preferable depending on the structures inherent in the population.

- The size of the sample refers to the number of subjects that are surveyed. In my case the subjects are people who are expert in either RE or SPI, or both.

The variability in responses decreases as the sample size increases (Fink 1995).

- Sampling activities include following up on eligible people who fail to respond to the questionnaire. Yet, there are no agreed-upon standards for what constitutes a good questionnaire response rate (SPSS 1996). However, when the response is less than 50%, the issue of concern is whether the sample is representative of the population (Oppenheim 2001).

Therefore the size of the sample is just one of several factors to consider in designing a reliable survey.

According to Fink (1995), before considering the size of the sample it is important to decide on the objectives, questions or hypotheses that the survey is to answer. In my case the objective of the questionnaire is to test seven success criteria of the RE process model.

This completes the section on data collection methods. The following section shows how data collected has been analysed in the 3 studies.

3.6 Qualitative data analysis

Analysis of qualitative data can be and should be done using explicit, systematic and reproducible methods (Greenhalgh and Taylor 1997). According to Tesch (1990), there is no single method of analysis that can be used for all types of interview data. Further, there is no rigid format to analysing qualitative data as the process can be eclectic, containing several analysis procedures (Creswell 1994).

3.6.1 Grounded theory

The semi-structured nature of the focus group interview where all discussion has been recorded in full allows for a 'grounded theory' approach (Burnard 1991). Grounded theory is described as the discovery of theory from data through the process of constant comparison (Glaser and Strauss 1967). The aim of this method is to systematically record themes and issues addressed in the focus groups and to link the themes and issues together under a reasonably exhaustive category system (Burnard 1991). According to Burnard (1991), to carry out an analysis of the data using a grounded theory approach the researcher must work with the following assumptions:

- The identified common themes in the interviews are really 'common'
- One person's view can be linked with another person's

According to Shipman (1997), problems can arise from a grounded theory approach due to its priority to advance knowledge through concentration on the 'theoretically interesting'. Furthermore, observation and interpretation are so intertwined that the researcher may see a very different reality to those who are participating. It is possible that "meanings have been imposed rather than detected" which may lead to an exploitation of those researched (Shipman 1997).

3.6.2 Content analysis

Content analysis is a classification scheme that can be used with focus group data. The process of data analysis as described by Krippendorff (1980) is similar to the grounded theory method, where replicable and valid inferences are made from the data to their context. Where content analysis differs from grounded theory is that it is largely numeric and therefore includes a quantitative form of research. Although conclusions drawn from content analysis are not statistical; they are substantive (Tesch 1990). Classical content analysis is an "objective, systematic, and quantitative description of the manifest content of communication" (Berelson, 1952 in Tesch 1990).

According to Burnard (1991) there are 14 systematic stages involved in classifying focus group transcript data. These stages are adapted from a grounded theory and content analysis approach to the data and form the guidelines given in the Table 5.

Table 5: A qualitative approach to classifying focus group data

Stage	Guideline
1	Notes are made after the interview recording topics talked about in the interview
2	Transcripts are read through and notes are made on general themes within the transcripts
3	Transcripts are read through again and as many headings as necessary are written down to describe aspects of content. Open-coding is made where categories are freely generated. This often leads to many detailed categories.
4	The list of categories is surveyed and grouped together under higher order headings. The aim is to reduce the numbers of categories by 'collapsing' some of the categories that are similar into broader categories.
5	The new list of categories is again refined to remove any repetitions or similar headings
6	Two colleagues are invited to generate category systems, independently without seeing the researcher's list. This list is discussed with the aim of enhancing validity of the categorising method and guard against researcher bias
7	Transcripts are re-read alongside the finally agreed list of categories and sub-headings to establish the degree to which the categories cover all aspects of the interview. Adjustments are made as necessary.
8	Each transcript is worked through with the list of categories and sub-headings and is 'coded' according to the list of category headings.
9	Each coded section of the interview is cut out of the transcript and all items of each code are collected together. Retaining the associated quote gives the code a context.
10	These cut sections of the transcripts are combined with the associated category headings and sub-headings
11	Interviewees are asked whether the quote and category associations are appropriate. Adjustments are made as necessary
12	The findings are filed together and written up. Copies of the complete interviews are kept to hand.
13	Once all the sections are together, the writing up process begins
14	The researcher must assess what parts of the transcript to include; whether to use verbatim examples of interviews to illustrate the various sections, or just reference the text.

3.6.3 Qualitative approach to data analysis used in this study

The data collection process described in Table 5 is used in the studies of SPI problems and RE problems in chapters four and five respectively. Although researchers should plan their form of data analysis prior to data collection, this was not possible as the focus groups had already been conducted prior to my involvement in the project. A qualitative grounded theory approach as defined by Glaser and Strauss (1967) and applied by Burnard (1991) is therefore ideal as I need to familiarise myself with the data prior to creating any theory about the problems practitioners were experiencing with their SPI activities. Looking at the 14 guidelines given in Table 5, I was able to emulate all the stages with the exception of no. 1 and no. 11.

I used the broad principles of content analysis of each focus group discussion to develop problem categories by placing emphasis on the meaning, the interpretation and the intentions of the data, as offered by practitioners (Krippendorff 1980). Clusters of software development problems were identified. Chapters four and five present clear definitions of these problem groups, as in any data analysis the definition must be useful, simple and direct. By providing similar levels of granularity and clear detailed definitions I aim to "assemble trustworthy collections of software data from different sources" (Kitchenham *et al.* 2001).

Using the same focus group data, the second study is only concerned with RE process issues which are abstracted at a simple level. At this stage of analysis, I do not try to interpret why some problems were occurring but instead classify problems at the level of their occurrence, for example when a practitioner complains of a problem with requirements growth this is added and combined with all the other occurrences of this problem regardless of the underlying reason for that growth. Classification of the issues identified in the focus groups is presented in subsequent chapters of this thesis.

3.6.4 Cohen's kappa measure of agreement for problem classification

According to Silverman, a crucial requirement in content analysis is that the categories are sufficiency precise to allow different coders to arrive at the same results when the same body of material is examined (Silverman 1993). In any classification scheme it is essential that there is a common understanding of what each group represents to create data that is trustworthy; allowing the study to be extended, replicated and compared (Kitchenham *et al.* 2001). Indeed, the replication of previous studies is essential for the cumulative development of empirically grounded knowledge (Hakim 1987).

To add rigour to any conclusions drawn from the frequencies of focus group data a 'reasonable' level of confidence in the defined categories is required. Confidence in the subjective classification is gained through conducting a formal reliability test on the categories derived from problem quotes. Cohen's kappa (k) statistic is used to test this reliability. According to SPSS (2001):

"Cohen's kappa measures the agreement between the evaluations of two raters when both are rating the same object. A value of 1 indicates perfect agreement. A value of 0 indicates that agreement is no better than chance. Kappa is only available for tables in which both variables use the same category values and both variables have the same number of categories".

Landis and Koch (1977) provided some arbitrary benchmarks for the evaluation of observed *k* values. These benchmarks are as shown in Table 6.

Table 6: Cohen's kappa agreement benchmarks

K	*Strength of agreement*
0.00	Poor
0.01 – 0.20	Slight
0.21 – 0.40	Fair
0.41 – 0.60	Moderate
0.61 – 0.80	Substantial
0.81 – 1.00	Almost perfect

Dunn (1989) suggests, however, that the benchmarks given in Table 6 (Landis and Koch 1977) are 'too generous'. El Emam (1999) disagrees, stating that the benchmark is unrealistic and too stringent when applying them to process assessments in software engineering and calls for the development of a benchmark specifically for this field. This is especially relevant as many studies of interrater agreement of software process assessments have used this arbitrary benchmark to interpret results, for example (El Emam, Briand, and Smith 1996; Fusaro, El Emam and Smith 1997a; Simon et al, 1997, all in El Emam 1999).

However, although I use the *k* statistic in a software engineering context, I am not using it to test agreement in process assessments. The Cohen Kappa inter-rater reliability test is used to test the reliability of my classification scheme. The process of calculating the agreement index between two independent researchers is described as follows:

- Researcher 1 performs a manual analysis on practitioner responses to researcher questions on problems they were experiencing with SPI. All problems cited by each practitioner group are identified. This list of all the problems is then organised into problem groups.

- Researcher 1 develops the scale, and performs an initial analysis. Quotes from the transcripts are selected that represent a subset of all problems.

- Researcher 1 prepares a subset of quotes from the transcripts and presents these quotes to Researcher 2 along with definitions of problem classifications. (The quotes and classifications are separated)

- Researcher 2 places the quotes into given categories. Researcher 2 returns the classification to researcher 1 and discusses any ambiguity.

- Researcher 1 performs a Cohen's kappa inter-rater reliability test where the results from researcher 1 and researcher 2 are compared. The analysis is performed using SPSS version 11.0.

- The resulting k statistic is then compared to values given by the Landis and Koch (1977) benchmarks in Table 6.

Dunn (1989) acknowledges that any series of standards such as these are bound to be subjective. It would appear that there is no simple answer to the question, 'How good is an agreement?', but the Cohen Kappa statistic can be used, with caution, to add confidence to the reliability of my classification schemes.

3.7 Quantitative data analysis

3.7.1 Parametric and nonparametric methods

Parametric methods are procedures for testing hypotheses about parameters in a population described by a specified distributional form which is often, but not always 'a normal distribution'. Nonparametric methods or 'distribution free' methods, on the other hand, are based on a function of the sample observations, the probability distribution of which does not depend on a complete specification of the probability distribution of the population from which the sample was drawn. Consequently nonparametric techniques are valid under relatively general assumptions about the underlying population (Sprent 1993).

One of the unresolved issues in statistics is the question of when parametric rather than nonparametric tests should be used to analyse data. Some authors suggest that violation of the assumptions generally has little effect on the value of parametric tests – although there are a few exceptions to this rule (Cramer 1997). Despite the apparent robustness of these parametric test conditions, the statistical methods of analysis used in this thesis are predominantly nonparametric. As noted by Kitchenham et al (2002b) most nonparametric methods "are very efficient relative to their parametric counterparts and they are effective with small sample sizes". This is echoed by Sprent (1993) who states that in many cases nonparametric tests are only marginally less powerful than their parametric analogues and "nonparametric methods are often the only ones available for data that simply specify order, ranks or counts in various categories".

I have, however, applied the parametric confidence interval statistic in a similar way to Cramer (1997) and El Emam and Jung (2001) who also have converted nominal data to counts. These counts, which although discrete (not able to be represented as fractions), do have ratio properties. Anything that has ratio properties has also interval properties. Black (1999) also advocates this approach.

3.7.2 Frequency data analysis

The first step to organizing raw data is to group the data into independent categories and define these categories. These categories should be independent (ie. non-overlapping) and mutually exclusive (ie. every value will fall into one, and only one, category). The data are then presented as scores or values in a frequency or contingency table. Contingency tables arise when observations on a number of categorical variables are cross-classified. "Entries in each cell are the number of individuals with the corresponding combination of variable values" (Everitt 1998).

Typically the categories used to construct contingency tables are either nominal or ordinal (Black 1999). In most cases, the column and row categories in the contingency tables in this study are qualitative and nominal, but the counts/frequencies can be described as quantitative/ratio.

3.7.3 Chi-Square

Having created contingency tables that categorise all the identified software development problems, practitioner groups and CMM levels, there are several tests that can be used to interpret the data. For example the Pearson chi-squared test (χ^2) helps to determine the independence of the variables. If the variables are not associated they are said to be statistically independent. The χ^2 test compares the observed frequency of cases against the expected frequency assuming that the row and column variables are independent (Cramer 1997). If the resulting p-value is very small (conventionally, $p<0.05$), then it is unlikely that the observed test statistic occurred by chance, and so it is assumed that the null hypothesis is in fact false. (The *p*-value shows the probability that an observed result is due to chance rather than to participation in a program.)

In my case I look to results where $p < 0.05$, indicating that there is a less than 5% probability that the results are due to chance. If the p-value is small the variables are said to be associated. As an inferential statistic, the chi-squared test allows me to draw conclusions about the population on the basis of my sample results. For example, I can determine whether an apparent association between how senior managers and project managers view the RE process is the result of a real association between practitioners and RE process issues.

However, there are certain conditions associated with using the chi square to test the independence of variables as given by Miller et al (2002):

(a) For a 2 by 2 table chi-square should not be used if any of the expected frequencies are less than 5.

(b) For tables larger than 2 by 2 chi square should not be used if any of the expected frequencies are less that 1 or more than 20% of the expected frequencies are less than 5.

The data in some of my finer-grained analysis contain some low observed frequencies that may result in expected frequencies that are less than 5. I therefore used the SPSS (2001) Crosstabs procedure to automatically check that the data complied with the conditions cited in (a) and (b) above. Whenever the expected frequencies did not comply to these recommendations also given by (Cochran 1954 and Seigel 1956 in (Cramer 1997)) I used the nonparametric 'exact test' (SPSS 2001), as a means of obtaining accurate results. This nonparametric test enabled me to obtain an accurate significance level, as "the exact significance is always reliable, regardless of the size, distribution, sparseness, or balance of the data" (SPSS 2001).

The chi-squared test indicates whether there is a significant association between two variables, but does not give a measure of that association. When a significant association between variables results from the chi square test, I am interested in obtaining a visual display of the pattern of relationships among the categories of the variables. In order to gain a deeper understanding of these patterns and gain a measure of association between variables, I use the descriptive 'correspondence analysis' statistic to give me a measure of association.

3.7.4 Correspondence analysis

Everitt (1998) describes correspondence analysis (CA) as a method for displaying the relationships between categorical variables in a type of scatterplot diagram. For two such variables displayed in the form of a contingency table, a set of coordinate values representing the row and column categories are derived. These coordinate values are then used to allow the table to be displayed graphically. Such an analysis allows a visual examination of any structure or pattern in the data. Euclidean distances approximate chi-squared distances between row and column categories (Everitt 1998).

CA has been used in the social sciences to display descriptive category associations, see for example (Denzin and Lincoln 2000). CA is a multivariate statistical method used to explore contingency table data by converting nominal data counts into graphical displays, called 'maps' (Greenacre and Blasius 1994). It is an exploratory technique used to reveal associations in the data. Data in a typical two-dimensional contingency table (both the row variables and the column variables) are represented in the same geometrical space. This means that relations among row or column variables and between row and column variables can be examined (Weller and Romney 1990).

To explain how contingency or frequency tables are converted to CA maps I use fictitious data relating TV viewing preference to age group. Frequencies from a two-way contingency table are converted to percentages to help compare the values in the tables, also marginal totals and mean averages have been added to aid understanding. However, in my study, the raw data in the form of counts are retained in the tables in order to present a full picture (Kitchenham *et al.* 2002b).

CA interprets the data by comparing the percentages against the mean average to draw associations between CMM levels. These percentages are examples of mathematical vectors that have a geometric interpretation as they define points in a multi-dimensional space. The fictitious data in Table 7 explains how the elements (percentages) are used as co-ordinates on a CA map. Each percentage is condensed into a unique point in this space and is called a 'profile'. The dimensionality is reduced so the profiles can be visualized in a more accessible 2-dimensional space.

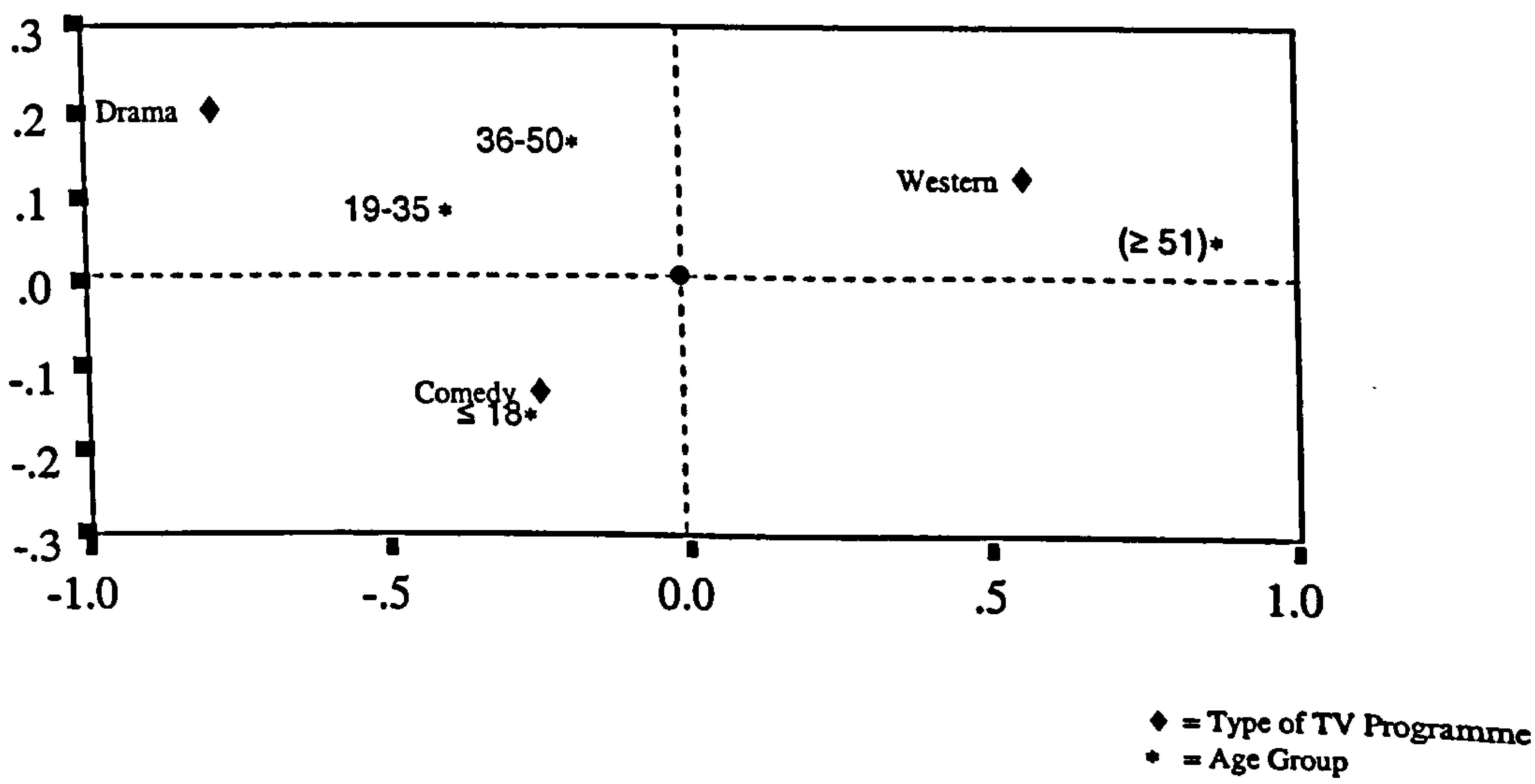
Table 7: TV programme preference by age group given in row profiles (in % form)

	Western	Comedy	Drama	Total
Age Group				
≤ 18	38	40	22	100
19-35	34	40	26	100
36-50	41	37	22	100
≥ 51	68	27	5	100
Total	181	144	75	400
Mean Average	45.25	36	18.75	

Data in Table 7 are converted from the 'euclidean' distance into a 'chi-square distance' as shown in Figure 4. This is to standardise any variance in frequencies, as:

"If no such standardization is performed the differences between larger proportions will tend to be large and thus dominate the distance calculation, while the small differences between the smaller proportions tend to be swamped. The weighting factors in the chi-square distance function thus tend to equalize the roles of the response options in measuring distances between the profiles" (Greenacre and Blasius 1994 pp 11-12).

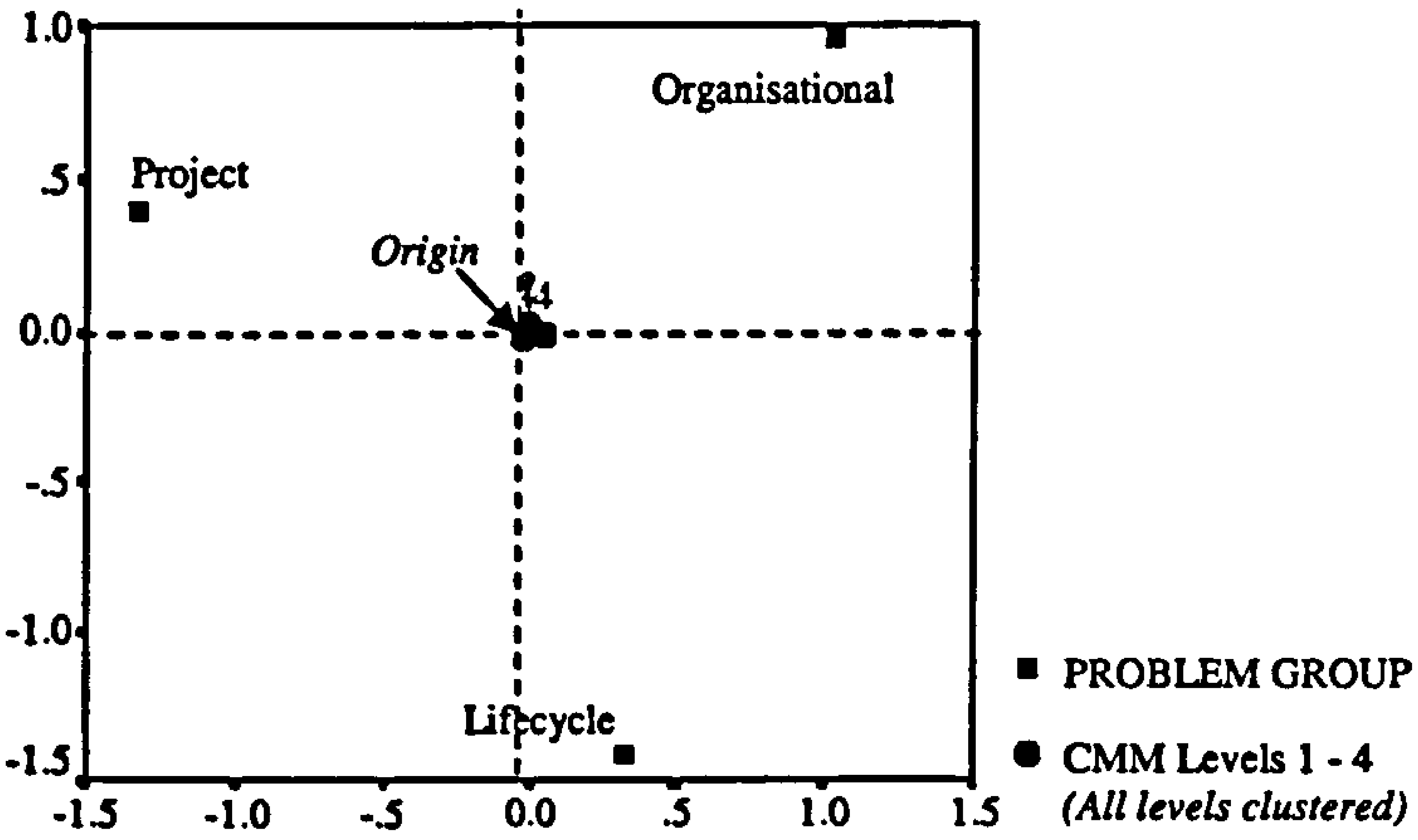
Figure 4: Correspondence Analysis Map showing
Age groups and their TV programme preferences



The CA map in Figure 4 shows, for example, that the 51 and over age group is most closely associated with Westerns and that the 18 and under age group is most closely associated with Comedies. It also shows that age groups 19-35 and 36-50 have similar TV programme preferences.

As CA takes account of the differences in the sample, it is an appropriate method for describing the focus group data where category variables are not equally represented. The data transformation in CA removes differences in magnitude among row and column totals, leaving the association or interaction. In other words, CA begins with a normalization of the data (Weller and Romney 1990).

Figure 5: Low Inertia: SW-CMM level vectors graphically display agreement



Total Inertia 0.001

3.8.4.1 Inertia

A measure of the distance (the chi square distance mentioned in the section above) between profiles against the average expected profile is called 'inertia'. A vector with a co-ordinate that is far from its 'centroid' (or average) will have a high inertia, while a vector with a co-ordinate near to the centroid will be near to 0 and will have a low inertia.

How to interpret levels of inertia can be explained by examining the two maps in Figures 5 and 6 below. Figure 5 shows a low level of inertia as all groups have similar responses and produce 'average' results. Figure 6 shows a high level of inertia where all groups have totally different profiles suggesting that there is an association between CMM level and problem group. I again use fictitious data to demonstrate these extremes.

- Low Inertia

Table 8: Row % of SW-CMM levels and problem groups with similar profiles

CMM Levels	PROBLEM GROUP			Active Margin
	Organis'l	Project	Lifecycle	
1	33	35	32	100
2	34	35	31	100
3	32	35	33	100
4	35	32	33	100
Active Margin	134	137	129	400
Mean Average	33.5	34.25	32.25	100

The percentages in Table 8 show that all four CMM Levels are reporting a similar number of problems in each of the given categories. This is further confirmed in a chi-squared test of association result where $p = 0.99$. The CA Map in Figure 5 uses these profiles to graphically represent how CMM levels relate to these problems. There is no significant difference between how each CMM level reports its problems.

A total inertia of .001 in Figure 5 shows a strong problem agreement. The CA Map expresses this measure of agreement through a low inertia where all CMM levels are clustered around the origin or 'centroid' (0).

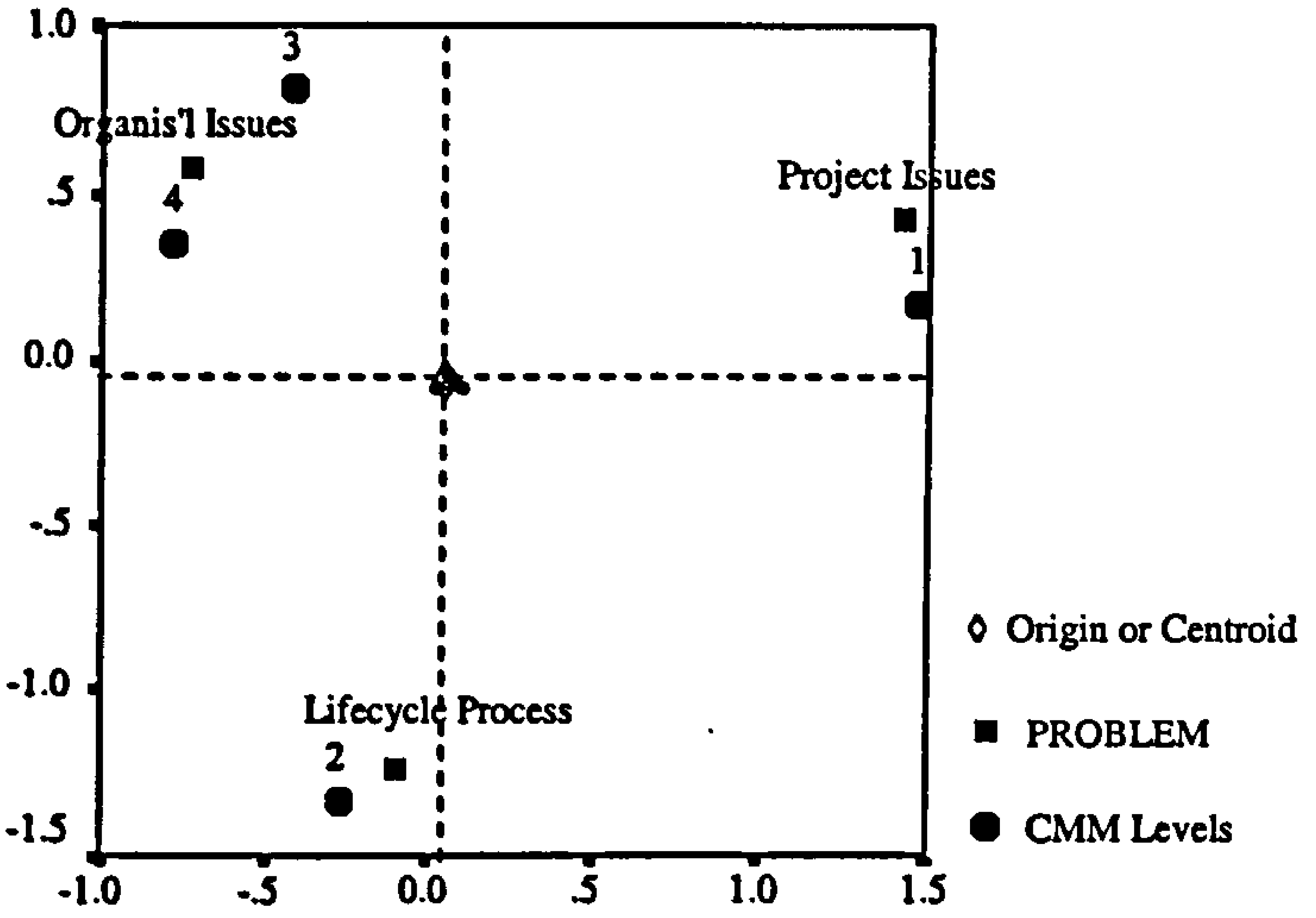
- High Inertia

In contrast, the fictitious row percentages presented in Table 9 show that CMM Levels have a significant association with the problem group. There is very little likelihood of these associations occurring by chance with a chi-squared test of independence of $p < 0.001$. Figure 6 graphically displays this extreme polarization of profiles with a total inertia close to its maximum. The total inertia of 0.999 shows a polarisation of problems.

Table 9: Row % of SW-CMM levels and Problem Groups with polarised profiles

CMM Levels	PROBLEM GROUP			Active Margin
	Organis'l	Project	Lifecycle	
1	1	80	19	100
2	19	1	80	100
3	80	19	1	100
4	80	1	19	100
Active Margin	180	101	119	400

Figure 6: High Inertia: SW-CMM level vectors graphically display polarisation



Total Inertia 0.999

3.7.5 Implementing correspondence analysis in this study

Correspondence analysis is used in this study to graphically display the relationship between the nominal variables in my frequency or contingency tables that are derived from the focus group data. This exploratory technique will help to describe the relationships 'within' variables (e.g. how practitioner groups relate to each other in how they perceive problems), as well as the relationship 'between' variables (e.g. how each practitioner group relates to each problem group).

3.7.6 Questionnaire analysis

The questionnaire used in this study is designed specifically to validate the RE process capability maturity model (R-CMM). The questionnaire data is mainly subjective; the only objective data collected being demographic. The questionnaire was designed to produce responses that allow for quantitative analyses of model success criteria. I have used some statistics in the analysis of this data in order to establish patterns in responses and highlight possible inconsistencies. My aim is to present reliable results and the findings that are derived from the use of statistics are reported in the individual studies. In this section I give a brief outline of the various statistics used and how they apply to the questionnaire data.

3.7.6.1 Mann-Whitney U test

A Mann-Whitney U test is used to compare how two groups of experts respond to a number of key items (Siegel and Castellan 1988). This nonparametric test compares the responses of two independent groups. For example, Dutta and van Wassenhove (1997) used the Mann-Whitney U test to confirm statistically significant differences between leading and lagging countries in Europe. The results showed that it is not always appropriate to treat all European countries as one uniform 'block' as is commonly done in many analyses.

I use the Mann-Whitney U test in SPSS (2001) to compare how SW-CMM critical experts and SW-CMM supportive experts view the R-CMM. According to SPSS, Mann-Whitney U "tests whether two independent samples are from the same population. It requires an ordinal level of measurement. U is the number of times a value in the first group precedes a value in the second group, when values are sorted in ascending order" (SPSS 2001).

I am particularly interested in whether positive or negative perceptions of the SW-CMM influence how the experts respond to R-CMM related questions. Indeed, if there is an association in how the expert views both models this should be taken into account when reporting the results. A discussion of how I have applied this test is given in the final study in this thesis.

2.7.6.2 Cohen's kappa measure of agreement

Cohen's kappa statistic is used to test the subjective classification of questionnaire items. Questionnaire responses are grouped together to gauge the level of support for the RE process model success criteria. However, as this classification of questionnaire responses is subjective, I need to test the reliability of my methods.

Questionnaire items are classified according to the seven success criteria. The Cohen Kappa interrater test is used to validate how well the items in the questionnaire actually link to the given success criteria. The process involved two researchers as follows:

- One researcher looks at the items in the questionnaire and places them into one of seven established categories which represent the success criteria.
- This practice is repeated by a second researcher unfamiliar with the work, to give a more objective view of this classification scheme.
- A comparison of the results is then made to show how two independent researchers group the questions.
- The reliability of these categorizations is tested through the Cohen's kappa (κ) statistic where agreement between the evaluations of two raters (rating the same object) is measured (SPSS 1999).

3.7.6.3 Confidence Intervals

As I designed the R-CMM I may be biased in my design of the survey instrument and how I evaluate the responses. I argue that the subjective design of the survey instrument is unavoidable and a limitation of a study involving a small group of people with limited time-scales and small resources. However I endeavour to counter the potential weakness in the evaluation of the survey by reporting all responses to the questionnaire as raw scores, prior to making any observations about these scores. The methodology explained in this section shows my approach to interpreting the response frequencies in the validation of the R-CMM.

The experts involved in this study are not a large group selected through a scientific sampling method, although I believe them to share many characteristics with the population of experts as a whole. Although it is possible to use the sample data to calculate the proportion of the sample ($\hat{p}$) which is supportive of each statement, and this sample value is the best estimate of the value of this proportion in the population, it is unlikely to be equal to the population value. Based on the calculation of the standard error of the sample statistic, I can place an interval around the sample statistic that specifies the likely range within which the population value is likely to fall. This interval is called a confidence interval (Newton and Rudestam 1999).

The term confidence interval refers to the degree of confidence, expressed as a percentage, that the interval contains the population proportion. A 95% confidence interval will contain the population value with 95% probability. This means that, on average, 5% of intervals constructed will not contain the population value. The width of the confidence interval (CI) is determined by the confidence level and the sample size, n, which is used in the calculation of the standard error of the estimate. CIs use a critical value (z value) from the standard normal distribution, corresponding to the confidence level. The higher the degree of confidence, the wider the confidence interval - I have chosen to construct 95% confidence intervals, as this is the most conventional value, analogous to carrying out significance tests at the 5% significance level. The larger the sample size, the smaller the standard error and the narrower the confidence interval.

The formula for the calculation of the confidence interval for a proportion used by El Emam and Jung (2001) is found in most standard statistics textbooks, for example, (Altman 1991; SPSS 1996). However, use of this formula requires the sample size n to be quite large, and/or the proportion $\hat{p}$ to be approximately equal to a half. A rule of thumb for the use of this formula states that $n\hat{p}$ and $n(1-\hat{p})$ must both be >5 (Altman 1991). This is unlikely to be the case for my data. As such, I will instead use the formula for the score confidence interval due to Wilson (1927) and given below

$$\left(\hat{p}+\frac{z_{\alpha/2}^2}{2n}\pm z_{\alpha/2}\sqrt{\left[\hat{p}(1-\hat{p})+z_{\alpha/2}^2/4n\right]/n}\right)/\left(1+z_{\alpha/2}^2/n\right)$$

where n = number of valid responses, $\hat{p}$ = proportion of supportive responses, α = 0.05 for a 95% CI, and $z(\alpha/2) = 1.96$ from tables of the standard normal distribution.

Agresti and Coull (1998) show that this formula, unlike that used by El Emam and Jung (2001), gives the desired level of confidence "for virtually all combinations of n and p".

Figure 7 shows how I interpret the responses to a question where 18 experts gave a valid response, and 94% of the responses were supportive. I exclude the 'no opinion' response category scores (as shown in Figure 7) from the number of valid responses as they are neither supportive nor critical of the model. El Emam and Birk (2000) also collected their data through a questionnaire; however they regarded the 'don't know' responses as missing values and proceeded to assign values to this response category using the method of multiple imputation. I do not consider this method appropriate for my study considering the small size of the sample. For this reason, I do not incorporate the 'don't know', 'no opinion', 'missing' or 'neutral' responses into the analysis of the strengths and weakness of the R-CMM.

Question	'No opinion' responses	No. of valid responses	Supportive responses		Confidence interval 95 %	
			Observed	%	LL	UL
21. How consistent is the level of detail given within the Requirements CMM?	2/20	18	17	(3) = 50 (4) Very = 44 Total = 94	[74,	99]

Figure 7: Reporting Confidence Intervals

- Benchmarks for action

El Emam and Jung (2001) assessed 80% supportive responses in the sample to be the threshold for taking action based on developer feedback. With my small sample size, I look to the confidence intervals (CIs) to guide me towards interpreting the true level of support given to each item. As I have used multiple items to address most of my success criteria, I use the results to compare, in relative terms, where the model's strengths and weaknesses are. Where the CI does not include 80% I take as an indication that more work is required in this area of the model. I use the lower bounds of the confidence limit to show agreement where model strengths are indicated by both the LL and UL being 80% or above. Using this criterion means that, in practice, 80% for both LL and UL can only be achieved with my data when there is 100% support.

3.7.6.4 McNemar's change test

The questionnaire is designed to gain an understanding of where strengths and weaknesses in the model occur through an analysis of multiple items. However in some cases, responses to similar questions appear to contradict each other. In such cases I would like to test whether there is a significant difference in how the subjects are responding to these questions. If there is a significant difference there may be a problem in how the questions are interpreted, or it may show that the experts are ambivalent about this area of the model. McNemar's change test is an ideal method as it reveals potential differences by comparing binary response patterns between 2 matched conditions. It may be thought of as a nonparametric equivalent to the paired t-test. It is appropriate to use this test to compare the responses to questions by each participant, providing the responses are dichotomised. I have achieved this by dividing responses into those that are supportive and those that are critical. This is consistent with previous questionnaire data interpretation as given in Figure 2.

Differences can be confirmed by carrying out McNemar's test in SPSS (2001) and given the relatively small sample size, I specify an exact test. According to Maxwell (1975) "the appropriate test for comparing frequencies in matched samples is one due to McNemar (1955)". The results of applying this test are given in the final study in this thesis.

This completes the section on analysis methods used in this thesis. This chapter now concludes with a summary of the methods used.

3.8 Summary of the methodology

In this chapter I have described how both qualitative and quantitative research methods are used in this thesis to collect and analyse the data. Some of these empirical methods, such as Correspondence Analysis are new to software engineering, while other methods such as the Cohen Kappa statistic are more common. I have shown why methods are used and how they help to provide a context for the data. The use of methods requires an understanding of the data as each method has associated assumptions. I do not claim that the methods used are necessarily the only methods suited to my investigation. Results are presented with an understanding that

"Experiments are done in the real world and are therefore never perfect. Any empirical study, and especially a novel one, has flaws" (Tichy 2000, page 1).

In line with Tichy's pragmatic approach to empirical research, my research can be characterized in terms of the questions I am investigating, the research methods I adopt and the criteria by which I evaluate the results (Shaw 2002).

I conclude this section with a caveat that the findings that result from these methods are not necessarily final and complete. However, as I provide evidence that methods are properly implemented, the combination of methods of collection and analysis should provide research with a sound basis for further work. As Shaw (2002) notes:

"Major results that influence practice rely on accumulation of evidence from many projects. Each individual paper thus provides incremental knowledge, and collections of related research projects and reports provide both confirming and cumulative evidence" (Shaw 2002, page 9).

PART TWO

REQUIREMENTS ENGINEERING PROBLEMS: A PROPOSAL

Chapter Four: Software development problems – an empirical study

4.1 Introduction

This chapter presents an empirical study of the problems twelve UK software companies experienced in software development. In total I present qualitative data collected from forty-five focus groups that involved over two hundred software staff. This study forms part of the PPP project that focuses on managing practitioners' impact on process and product. The problems highlighted in this study provide the initial motivation for developing a model to help guide practitioners in their requirements engineering activities.

Quantitative methods are used to assess whether there is an association between process maturity and the types of problems companies are reporting. A range of analysis methods are used to establish whether there is an association between practitioner groups and software process improvement problems. If there is an association, this has implications of how a model to support practitioners is applied. The methodology used includes content analysis and correspondence analysis.

4.2 Study aims

The study aims to answer three research questions through identifying the problems experienced by key software development staff at twelve software development companies. All the companies were involved in SPI and had an idea of how mature their processes were in terms of the SW-CMM model. The investigation involved three types of practitioners who were divided into groups of developers, project managers and senior managers.

The aim of this study is to answer the following research questions:

4.2.1 Research question 1: Is there an association between software development problems and SW-CMM levels?

My first research question focuses on whether there is an association between an improved RE process and higher process maturity levels. A growing number of companies are using the SW-CMM as a basis for improving their software processes (Paulk *et al.* 1995; SEI 2003b). Numerous studies report positive and negative factors that impact SPI (Herbsleb and Goldenson, 1996; El Emam *et al.*, 1998; Stelzer and Mellis, 1998; El Emam *et al.*, 1999; Dybå, 2000), but few if any relate problems occurring within the process to the company's current or targeted SW-CMM level. Because identifying and resolving problems is essential, and because of the lack of previous research relating problems to SW-CMM maturity levels, I investigate whether companies at different levels of maturity report different kinds of problems.

All companies in my sample have formally or informally assessed their process capabilities in accordance with the SW-CMM (see company profiles in Appendix B, and Company audits in Appendix C). If the study indicates that there is an association between software development problems and SW-CMM levels this would suggest that when developing a specialised model, I should retain this maturity concept.

4.2.2 Research question 2: Do developers, project managers and senior managers have different software development problems?

My second research question looks at whether the three practitioner groups have different problems with SPI. The importance of recognising different needs is highlighted in the literature that states that improving software processes is not the province of any one particular practitioner group (Baddoo 2002; Diaz and Sligo 1997; Krasner 1997). Stelzer and Mellis (1998) warn that unless companies openly involve staff at all stages during implementation of improvement programmes, investment and best efforts are wasted.

In my study the problems cited by senior managers, project managers and developers are examined separately to gain a staff perspective of SPI issues. The literature has shown that to achieve synergy necessitates actively looking for points of disagreement. It is therefore of practical use to highlight similarities and differences in problems practitioners are experiencing in their software improvement programmes. Developing an understanding of the problems associated with each role will help companies achieve a more open approach to SPI and will help in the design of a model that addresses the different needs.

If the three practitioner groups are reporting different development problems, this would suggest that in any software process improvement exercise, each practitioner group should be involved and represented. This has implications for the design and application of my specialised process maturity model.

4.2.3 Research question 3: How do requirements engineering process problems relate to other software development problems?

A purpose of this study is to place RE process problems in context with other software development problems. Although it is not possible to generalise from the results, due to the size and type of sample, I am interested in whether the sample reflects the literature in showing the RE process to be a major impediment to software development, and in turn, to software process improvement. Should RE processes prove to be a major problem, this justifies further research in this area.

This third research question summarises the results of the previous two elements of the study. The frame of reference is now focussed on RE process related issues. This study therefore aims to provide a context for RE related problems against the wider issues associated with software development.

4.3 Focus groups

From September 1999 to March 2000 researchers involved in the early stages of the PPP project visited the twelve software companies and conducted 45 focus groups as reported in this study. Focus groups were just one of the methods the group used to collect data, see for example Hall *et al.* (2002c). However, it was the most free-flowing of the methods and led to a rich and varied data collection that allowed a qualitative investigation into problems practitioners were experiencing in their daily software development activities. The PPP group were careful to create an environment where the subjects were able to talk freely amongst their own peer group, and were guaranteed anonymity.

I joined the PPP project in October 2000, shortly after the focus group data had been collected and therefore had no control over the data that was available. This also means that I was not able to dictate the way in which data should be collected. According to Fenton (2001) although this lack of control might be regarded as an impediment to carrying out research, in fact "it should be regarded as the norm". It could even be viewed as an advantage as I was able to view, analyse and interpret the data without any pre-conceptions. In the same way as Fenton carried out his research (Fenton 2001), I look at data that is available and retrospectively consider the most general and useful software engineering hypotheses that can be drawn from the data. By focussing on providing small pieces of evidence my work can support some of the software engineering hypotheses highlighted in the literature.

4.3.1 The participating companies

The sample of participating companies were drawn from a wider study of SPI activities in the UK (Hall *et al.* 2002c). This initial PPP group study, involved sending questionnaires to a sample of SPI managers identified using public domain information about software development companies. This information included relevant mailing lists and conference attendance lists. Questionnaires were mailed to SPI managers at one thousand companies and two hundred replies were received of which eighty were fully completed.

There are no agreed-upon standards for what constitutes a good questionnaire response rate (SPSS 1996). However, when the response is less than 50%, the issue of concern is whether the sample is representative of the population (Oppenheim 2001). An unsolicited mailing of this type often results in a response rate of less than 20% (SPSS 1996). Indeed, placing the sample in context, this low response can be expected considering the number of UK software companies with a formal SPI programme (Baddoo 2002). Despite the confidence shown in the data (Baddoo 2002), I exercise caution when making observations about the data as there is a likelihood of bias due to 80% of targeted companies not responding to the mailing.

4.3.2 The process by which the participating companies and focus group subjects were selected

Companies were selected from the eighty companies who completed the detailed questionnaire that included broad information about their software development activities and company demographics (Hakim 1987). Thirteen companies were specifically chosen to provide the research project with a cross-section of company maturity levels, software applications and company sizes. SPI managers were asked to select focus group participants by choosing every fourth person on the staff list. Although this quasi-random selection method was undertaken in many of the companies, not all managers adhered to this request. Therefore the reliability of participant representation is variable.

In my study, I use data collected from twelve companies. Appendix B provides a demographic overview of the companies where focus groups were implemented.

4.3.3 The process by which subjects were assigned to treatments

Focus groups were divided according to staff role as shown in Table 10. This is consistent with current best practice where holding separate sessions with homogeneous but contrasting groups is believed to produce information in greater depth than would be the case with heterogeneous groups (Shaw 1999).

Table 10: Focus groups by staff group

Company No.	Senior Managers	Project Managers	Developers	Total
1	1	2	2	5
2	1	1	2	4
3	1	3	2	6
4	1	3	2	6
5	1	1	1	3
6	0	1	2	3
7	1	1	0	2
8	1	1	1	3
9	1	0	2	3
10	1	2	2	5
11	0	1	1	2
12	Not used in this study			
13	0	1	2	3
Total	9	17	19	45

Twelve of the participating companies are represented in this study. They comprise forty-five focus groups (see Table 10 for breakdown of practitioner groups). Each focus group lasted approximately ninety minutes and included between four to six participants. Each session was audio-tape recorded and recordings were subsequently transcribed. All data has been anonymised and companies are referred to by numbers that are consistent across all PPP group publications.

4.4 Contextual framework

4.4.1 Practitioner groups

Multiple project manager and developer focus groups were conducted at eight out of the twelve companies (as shown in Table 10). This reflects the fact that these eight companies were considerably larger than the other four. Furthermore, it was not possible to assemble a group of senior managers at three companies for logistical reasons. Company 9 operates a flat company structure where there are no middle management or project management roles. Technical difficulties (a defective audio tape) prevented me from using data from the developer group in Company 7.

Each company, therefore, does not have a representative sample of three staff groups. So, to allow direct comparison between groups of different sizes, I have used methods that allow for these inconsistencies, e.g. Chi Square and Correspondence Analysis as explained in the next section. Also, as the sample is relatively small, the methods of analysis are adapted to allow for this limitation. These shortcomings are reported in the results section.

4.4.2 SW-CMM levels

Table 11: Companies involved in the study and their associated SW-CMM level

Company No	1	2	3	4	5	6	7	8	9	10	11	13
SW-CMM Level	1*	1	1	1	4*	3*	1	2	3	1	2	3

* Based on formal SW-CMM assessment. Companies without * are all undertaking SPI and have self-estimated their SW-CMM levels through answering questions in the questionnaire. (See Appendix C for further details of the self-assessment)

As Table 11 shows, the sample contains six companies at the lowest SW-CMM level (level 1), which accounts for 50% of the companies represented. This is no surprise as, according to Paulk and Chrissis (2000), an estimated 70% of software companies remain at this level. In their survey of high maturity organisations, Paulk and Chrissis refer to only 44 level 4 organisations and 27 level 5 organisations in the world (though they acknowledge there may be more).

Fewer UK companies than US companies have so far been formally CMM assessed and so it is consistent that only three of the companies in the study have been (SEI 2003b). To overcome this limitation the remaining nine companies were asked to estimate their CMM level through a detailed questionnaire. This procedure was also used by Herbsleb and Goldenson (1996).

As self-rated companies can over-estimate their process maturity I conducted my own independent study of the nine self-assessed companies' CMM Levels. Appendix C gives an overview of my methodology together with a detailed breakdown of company practices. This company audit confirms that the companies conform to their self-assessed maturity levels.

4.4.3 Perspective

The qualitative data collected in this research characterises practitioners' perceptions. These perceptions have not been verified directly. It could be that practitioners censor their comments to look good in front of their peers. Also, members of the focus group may be concerned about how the data might be used against them, and more subtle and less conscious fears may have affected their responses.

The results I present are perceptions from the development organisation. Contributions from users and customers who may perceive quite different problems were not elicited.

4.4.4 Problem generalisation

Although the focus group data provides interesting insights into the problems encountered in these twelve companies, it is not appropriate to generalise from this sample; e.g. there is only one company representing SW-CMM level 4, and there is some bias in how subjects were chosen. Ideally, I should have both more case studies (to accumulate evidence) and repeatable and controlled experiments (to determine the underlying causal factors) before I can consider my results definitive.

It is likely that each company undertaking SPI has individual and possibly unique problems. I acknowledge that companies are likely to vary in where their process problems lie and how they approach improving them. However, the companies involved in this study represent a mix of company maturity levels, practitioner groups, software applications, company sizes and project sizes. This cross-section of experiences is therefore helpful in gaining an impression of the type of SPI problems occurring in some UK companies.

4.4.5 Problem status

Finally, I make no comment on the importance of individual SPI problems cited. For my analysis every SPI process problem mentioned in a focus group has equal importance. Clearly in the commercial world some problems are more important than others.

4.5 Qualitative data analysis

I examined the PPP focus group transcripts and identified general focus group questions that related to problems practitioners were experiencing in their SPI programmes. I made a detailed analysis of the problems each focus group reported through a combination of grounded theory and content analysis as given in (Burnard 1991).

In order to investigate problems practitioners are experiencing in their SPI programmes, I examined responses to the following two questions asked to each focus group:

- What are the problems and issues in software development in your company?
- What are the obstacles to SPI in your company?

Taking a grounded theory approach to the data, I analysed over 1000 pages of focus group transcripts. Manually reading through these transcripts resulted in identifying 1252 problems. Theories grew out of the data, where despite the questions put to the practitioners being of a general nature, 'requirements engineering' process problems emerged as a recurring theme. It was through this grounded theory approach that the three research questions that frame this study were formed.

I then used the broad principles of content analysis in each focus group discussion to develop problem categories by placing emphasis on the meaning, the interpretation and the intentions of the data, as offered by practitioners (Krippendorff 1980). Clusters of software development problems were identified. I categorised each problem into three main groups and 16 sub-categories as in the scheme shown in Table 12.

Table 12: Three identified problem groups

Organisational Issues	Project Issues	Life Cycle Issues
Change Management	Budget and estimates	Requirements
Communication	Documentation	Design
Culture	Quality	Coding
Goals	Timescales	Testing
People	Tools & Technology	Maintenance
Politics		

Definitions of these problem groups are provided in Appendix I.

The coding scheme given in Figure 8 is an example of how qualitative data is categorised. The figure also shows how coding is used to allow the data to be traced back to its origin. During this process I endeavour to suspend any prior theoretical notions. Having grouped the quotes into these categories I now need to test the reliability of my classification scheme.

Figure 8: Problem Table example

A small section of a problem table is given as an example - quotes are fairly long to provide necessary context

categories and finer-grained sub-categories grew from the data

Quotes are taken directly from transcripts

For traceability - each quote is coded to link back to transcript: company, focus group and page

ORGANISATIONAL ISSUES (O)

Problem Group	Quote	Co/Staff Group/Page
1. COMMUNICATION		
a) Internal	We don't talk to the modeling department, we don't always talk to hardware department, we don't talk to systems. It is the interfaces, communication interfaces that I think is the biggest problem.	B/DevB/p.1
	historically we have always kept very separate.	B/DevB/p.1
	with big projects, like you have got 20 hardware engineers and 30 software engineers, that is more difficult to locate within one area. Communication does, sometimes, seize because of the control aspect.	B/DevB/p.1
	I think in our team(SPI team) all we work at is a sounding board for the management team.	B/DevB/p10
	Management Aims?: [Same as Dev] But we are looking at it at different levels.	B/DevB/p12
	there are other departments producing software who are not as aware as we are.	B/PMB/p.4
	we don't see enough of the other departments to know how far they are going at the moment, to be honest. They are doing estimates and things like that, but it is how they do it. [cut off]	B/PMB/p.6
	I will have to say that when I suggest improvements sometimes I don't get response back.	B/PMB/p.8
	We also think that it is important to give feedback as soon as possible to the people giving you ideas. Without that people are going to think, "well what is the point in giving you any idea?"… It is quite often that you give an idea an we whack it through in two weeks and the engineer who gave the idea is on a different project on a different part of a life cycle and is not benefiting from that idea, the department as an whole is, but the individual isn't.	B/DevB/p.13
	We are trying to institute processes to try and iron these things out [cost cutting, focus on SPI], but at the end of the day we get driven by the bits from other departments within the company	B/DevG/p.3
	[main obstacle of SPI implementation] Trying to explain why we are doing it to all disciplines. Explaining the benefits in a convincing way to other departments…. Also making engineers understand why it should be done.	B/DevG/p.12
	there is systems engineering activity which is attempting to do the same things but they are way behind, you know. So really until they catch up, it does make life difficult for the others. Software engineers are finding it difficult that the systems business isn't properly organised.	B/DevG/p.2

4.5.1 Validation of classification scheme

I conducted a formal reliability test, using the results from 2 different researchers. A Cohen's kappa inter-rater reliability test was performed using SPSS version 11.0. Cohen's kappa measure of agreement between the evaluations of the 2 raters was positive: 82 valid cases gave a value of k = 0.71 representing a "substantial" agreement (Dunn 1989; Vogt 1999). The 82 cases were selected from 16 focus group transcripts and related to three problem sub-categories as classified by me.

The k value of .71 is taken to indicate that the problem classification is reliable.

4.6 Quantitative data analysis

The result of my inter-rater reliability test, against the suggested benchmarks, is reliable (Landis and Koch 1977; Vogt 1999). Performing this test therefore adds rigour to my classification scheme and gives me confidence to proceed to the next stage of data analysis. I now draw up contingency tables based on 'reliable' problem frequencies. This entailed constructing a matrix that mapped all 16 sub-categories to each company, CMM level and practitioner group. The matrices included all 1251 problems abstracted from the transcripts. Individual contingency tables are drawn up based on the frequencies of problems within each group of interest. Examples of these contingency tables are given in Appendix G.

This process transforms the qualitative data into quantitative data and allows some statistical analyses to be performed on the data (Seaman 1999). It is only when I have drawn up frequency tables that classify all the problems identified by the practitioners in the focus groups that I inspect the data to look for patterns and trends (Tesch 1990).

The following sections 4.7, 4.8 and 4.9 present the results that relate to the three research questions.

4.7 SW-CMM level problem association

Research question 1: Is there an association between software development problems and CMM maturity levels?

Frequencies of all reported problems from the 45 focus groups have been converted to percentages to allow comparison (Table 13). For contingency tables showing raw data frequencies of all 12 companies by SW-CMM level see Tables 38 and 39 in Appendix G.

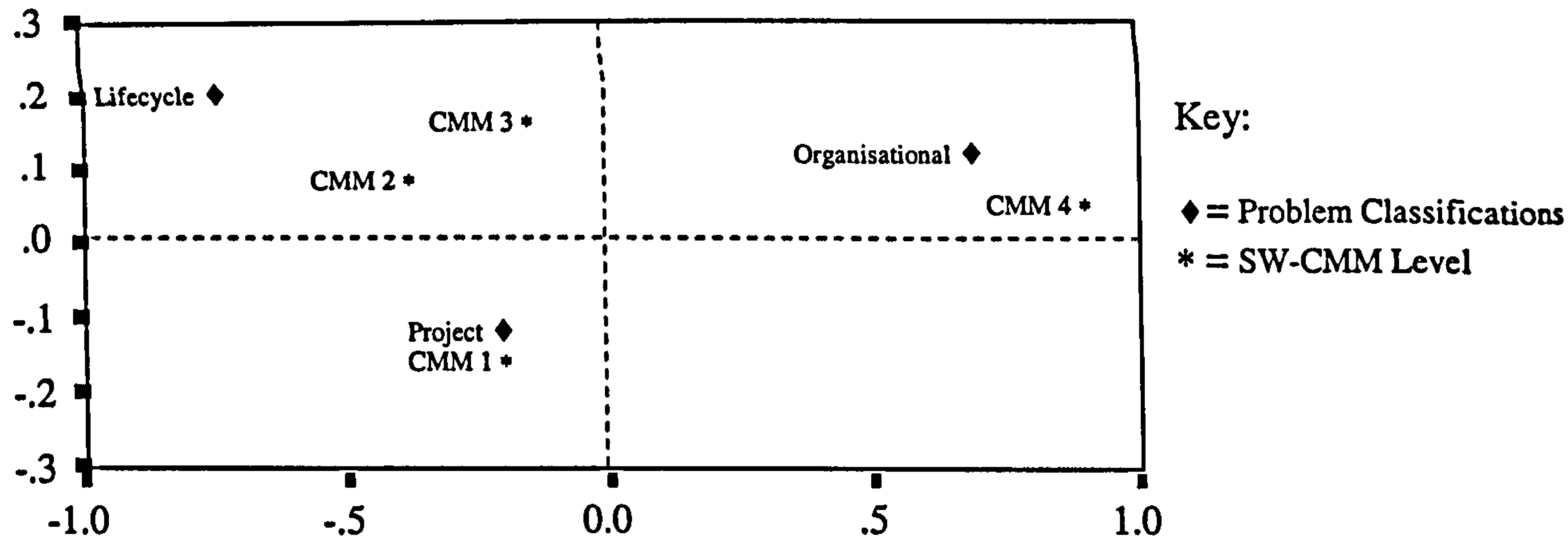
Table 13: SW-CMM levels and overall problem frequencies by row %

CMM Levels	Organisational	Project	Lifecycle
1	38	40	22
2	34	40	26
3	41	37	22
4	68	27	5

The significance of the relationship between the SW-CMM group and problem groups in Table 13 is confirmed by the chi-squared test of association $X^2 = 32.9$, df = 6, $p < 0.001$. It can be concluded that it is highly unlikely that the association between the four CMM levels and the three main problem groups appear by chance. In particular, the frequencies appear to show a strong relationship between the level 4 group and organisational issues (68%), and a gradual distancing from project issues as companies mature.

The percentages in Table 13 are used to create the correspondence analysis map in Figure 9. Figure 9 graphically shows the relationships between 'CMM levels' and 'problem groups', as well as within each group.

Figure 9: CA map of SW-CMM levels and SPI problem association



Total Inertia: 0.082

Although using the data from the four CMM levels shows a significant association, this is due to the level 4 company data. Omitting the level 4 variable from the analysis gives different results, with a chi-squared test of association $X^2 = 1.266$, df = 4, $p > 0.001$. There is no longer any association between the problem and CMM level variables and there is no sequential progression as seen in the CA map. In any case, all graphical displays in CA should be treated with caution.

4.7.1 Observations drawn from the data

The positions of the problem groups and the SW-CMM Levels in Figure 9, suggest the following:

a) As companies mature their concerns about project level problems weakens revealing a change in problem focus. Note in Figure 9 that:

- The SW-CMM Level 1 group is located very close to the project group. This suggests that the CMM Level 1 group is particularly concerned with problems at the project level.
- SW-CMM Level 1, Level 2 and Level 3 groups, and project issues, are all relatively close to the origin. This suggests that the three SW-CMM groups all share a 'relative' interest in problems at the project level.
- The SW-CMM Level 4 group is located much further away from the other SW-CMM groups or the project group. This suggests that the CMM Level 4 group has below average interest in problems at the project level.

b) As companies mature, they record more lifecycle problems but then move on to organisational problems. Note in Figure 9 that:

- The SW-CMM Level 2 and Level 3 groups are the groups nearest to the lifecycle group and are placed in the same quadrant. This suggests that these two groups suffer more lifecycle problems than levels 1 and 4.
- The CMM Level 4 group is located furthest away from the lifecycle group, suggesting that they are less burdened with lifecycle issues than the other 3 levels.

c) The highest maturity group appears to be the most different. The inertia of the SW-CMM Level 4 group is relatively high at 0.059. By contrast, the inertia of the other three groups are 0.005 (Level 1), 0.015 (Level 2) and 0.002 (Level 3). This suggests that the SW-CMM Level 4 group has the most distinct and unique perspective on software development problems.

d) As companies mature through the intermediate levels of maturity they 'move' closer together. In Figure 9, note that:

- The SW-CMM Level 2 and CMM Level 3 groups are the most closely located, and are within the same quadrant of the figure.
- By contrast, the SW-CMM Level 1 and SW-CMM Level 4 groups are further away from the Level 2 and Level 3 groups, and they are also in their own quadrants.

- This observation is consistent with Hayes and Zubrow's finding that it seems to be easier to mature from Level 2 to Level 3 than from Level 1 to Level 2. This may be because the two levels are closer in concept (Hayes and Zubrow 1995).

4.7.1.1 Differences between how Table 13 and CA map Figure 9 represents the data

The correspondence analysis map provides a different perspective on problem association, as shown in the following examples:

- Percentages in Table 13 show that CMM Level 1 and 2 companies share the same concerns with project issues (40%). This similarity is not shown in Figure 9 where a measure of association is given taking account of all variables. As SW-CMM level 2 companies have more concern with lifecycle issues than CMM level 1 companies, they are pulled away from project issues and move nearer to the lifecycle issues.
- Table 13 indicates that level 2 and level 3 groups have different problem profiles. Level 2 is most concerned with project issues while the main problems for level 3 are connected to organisational issues. However, the correspondence analysis identifies a similar differential between project and lifecycle problems which is reflected by the proximity of level 2 and 3 on the CA map.

4.8 Practitioner group problem association

Research question 2: Do developers, project managers and senior managers have different problems with SPI?

Table 14: Overview of problem classifications by practitioner group

	Practitioner Groups									Total		
	Dev (19 groups)			*PM (17 Groups)*			*SM (9 Groups)*			*(45 Practitioner Groups)*		
Problems	Frq	% col	% row	Frq	% col	% row	Frq	% col	% row	Frq	% col	% row
Organisational Issues	247	39.5	49.9	153	36.5	31.5	95	46.5	19.2	495	39.7	100.0
Project Issues	230	36.5	46.8	185	44.0	37.7	76	37	15.5	491	39.2	100.0
Lifecycle	151	24.0	57.0	80	19.5	30.2	34	16.5	12.8	265	21.1	100.0
Total	628	100	50.2	418	100	33.4	205	100	16.4	1251	100	100.0

Table 14 is a high-level abstraction of practitioners' experience in three main problem categories. The differences between staff groups and problems at this level of abstraction are significant with a chi-squared test of association result (using observed frequencies), $X^2 = 12.635$, df = 4, p = 0.013.

The CA map in Figure 10 takes the row percentages from Table 14 and converts them into a graphical view of the data.

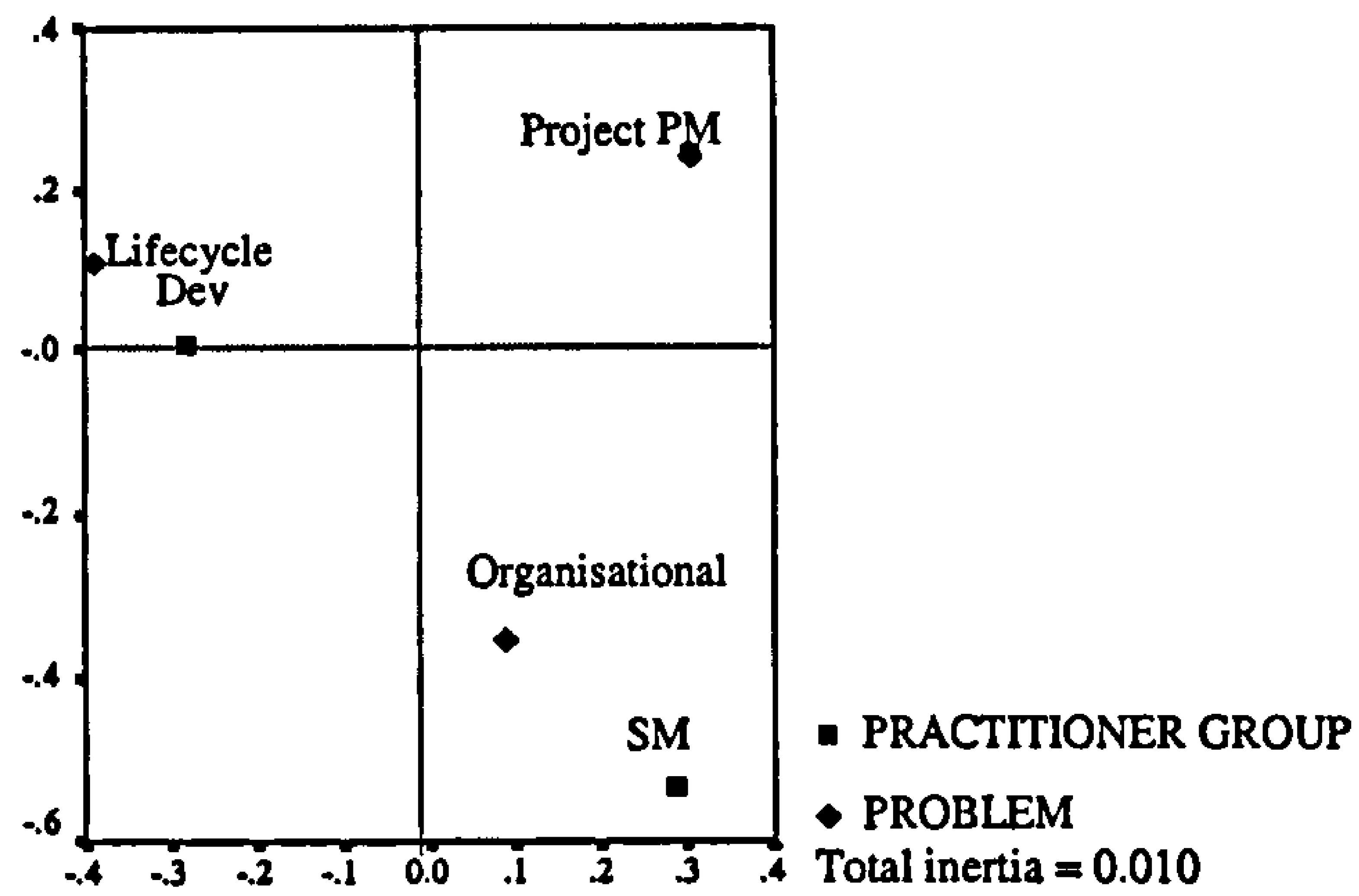


Figure 10: CA map of practitioner group and SPI problem association

Table 15: Top six problems by staff group
(data taken from 16 sub-category problem groups in Table 39 in Appendix G)

Problem Cited	Developers		Project Managers		Senior Manager		Total	
	Ranking	Col %	Ranking	Col %	Ranking	Col %	Ranking	Col %
People	1.5	13%	1.5	12%	1	16%	1	13%
Tools/Technology	1.5	13%	1.5	12%	5.5	9%	2	12%
Documentation	3.5	12%	3	11%	5.5	9%	3	11%
Communication	3.5	12%	5.5	9%	3	10%	4.5	10%
Requirements	5	11%	5.5	9%	3	10%	4.5	10%
Budget/Estimates	8	5%	4	10%	3	10%	6	8%
Total % (6/16 problems)		66		63		64		64

Figures given in Table 15 have been generalised across all CMM levels. The focus here is on the practitioner group and how they are citing RE process problems.

4.8.1 Observations drawn from the data

Taking the high level view presented in Table 14, the main findings are

a) Project managers are most concerned with project issues. Developers and senior managers are most concerned with organisational issues. (The CA map in Figure 10 shows these associations very clearly).

The graphical representation in Figure 10 gives confidence in the method used as it confirms expected relationships between problems and practitioner group. To gather more detailed information from the data and increase my knowledge of these relationships I have taken a finer-grained look at the 3 problem groups (organisational, project and software lifecycle).

b) Developers claim 57% of cited problems in the lifecycle group.

These observations are fairly superficial; in order to gain a more useful understanding of RE process problems as perceived by practitioners a finer grained analysis is undertaken.

Taking a more detailed view of RE problems as given in Table 15, and Tables 38 - 39 in Appendix G, the main findings are:

c) Most of the 16 sub-category problems cited are recognised by all practitioner groups. However, the frequencies of recorded problems vary significantly between practitioner groups. The likelihood of these relationships occurring by chance is minimal with a chi-squared test of association result of $X^2 = 137.52$, df $= 30$, $p < 0.001$. For raw data that includes all sixteen problem groups please see Table 39 in Appendix G.

d) There is a general consensus between all three practitioner groups as to the main problems they are experiencing in SPI (Table 15).

In all three-practitioner cases, the top six problems account for at least 63% of the total problems mentioned throughout the 16 sub-categories. Areas giving the greatest concern are People, Tools and Technology, Documentation, Communication and Requirements. Problems associated with people head the list of problems companies are experiencing in SPI. 'People' issues come under the umbrella of the 'Organisational' class and incorporate problems relating to:

Responsibilities, roles, rewards, expectations, blame;
Staff turnover, retention, recruitment;
Skills, experience;
Training.

For full definitions of all classifications refer to Appendix I.

e) The prevalent 'People' issues represent the most pressing problems for all groups and account for 17% of overall reported senior manager problems. Typical quotes are,

"[we are] very restricted in what we can do; answerable to 2 masters.. leads to conflicting directions"; "staff turnover in IT can be higher than 20% which causes instability"; "..knowledge is tied up with a few people"; "we have a lot of highly skilled people, but they are mainly isolated in their projects"; "training is poor...".

f) Developers and Project Managers appear to share how they view their problems in terms of their ranking of the top three problems. The 'Tools and Technology' category is recognised as a 'project' problem and is the second most mentioned problem for developers and project managers. It includes issues such as implementation of new technologies and tools and improvement methodologies such as SPI generally and the CMM specifically and pressures that inhibit the use of new tools. Developers typically state,

"Sometimes you don't have time to contribute to things like SPI on top of your day-to-day work"; "We are weak at technical infrastructure, for example we are still on Win 3.1". "Different departments decide they're going to move with different tools and there's no commonality"; "We spend a lot of time drawing a lot of pretty pictures, pretty graphs which no-one ever looks at".

g) Documentation is also high on the list of developer problems. This category includes co-ordination and management of documents, feedback, post-mortems and data collection methods. Developers report,

"There is no ownership of document production"; "There is no formal documentation"; "Documentation, we don't have enough, from a support point of view".

Project managers are also concerned with documentation and state that SW-CMM involves "too much paperwork. It is not as automated as it should be" (Project Manager, CMM level 1 company).

h) Senior Managers have below average concern for Project issues such as Documentation and Tools and Technology issues, as they concentrate on problems relating to People and Communication. They have above average concern for 'requirements engineering' issues in terms of problem ranking (equal 2nd), but an average concern in terms of percentage of problems. Indeed, further examination of Table 15 reveals that Developers devote a higher percentage of overall problems to RE processes than Senior Managers do with 11% and 10% respectively.

i) Developers do not share the high concern for 'Budgets and Estimates' with Senior Managers and to a lesser extent with Project Managers.

j) RE process issues, generally, are causing the greatest lifecycle problems.

4.9 RE process problems in context with SPI

Research question 3: How do RE process problems relate to other software development problems?

Within focus group discussions all development phases were identified as causing particular problems. Table 16 shows how problems experienced within development break down into requirements, design, coding, testing and maintenance.

Table 16: Development problems cited in 45 focus groups

	Frequency	Percentage
Requirements	130	49
Coding	16	6
Design	21	8
Testing	63	24
Maintenance	35	13
Total number of development problems	265	100

Table 16 shows that of all the development problems cited, 49% stem from RE processes. This is despite the fact that during data collection, the PPP study did not focus on the RE process and did not prompt participants in this direction at all. Indeed the following quote from a software developer indicates the significance of RE process problems reported in the focus groups:

"It is possible for us to start a project, get half way through it and the customer will turn around and say, this is now going to be used in a safety critical application ...".

A correspondence analysis was carried out to investigate the relationship between practitioner groups and how they view the finer-grained SPI problems (Figure 11).

Figure 11: Correspondence Analysis Map showing Measure of relationship between Practitioner Groups and Problem Groups

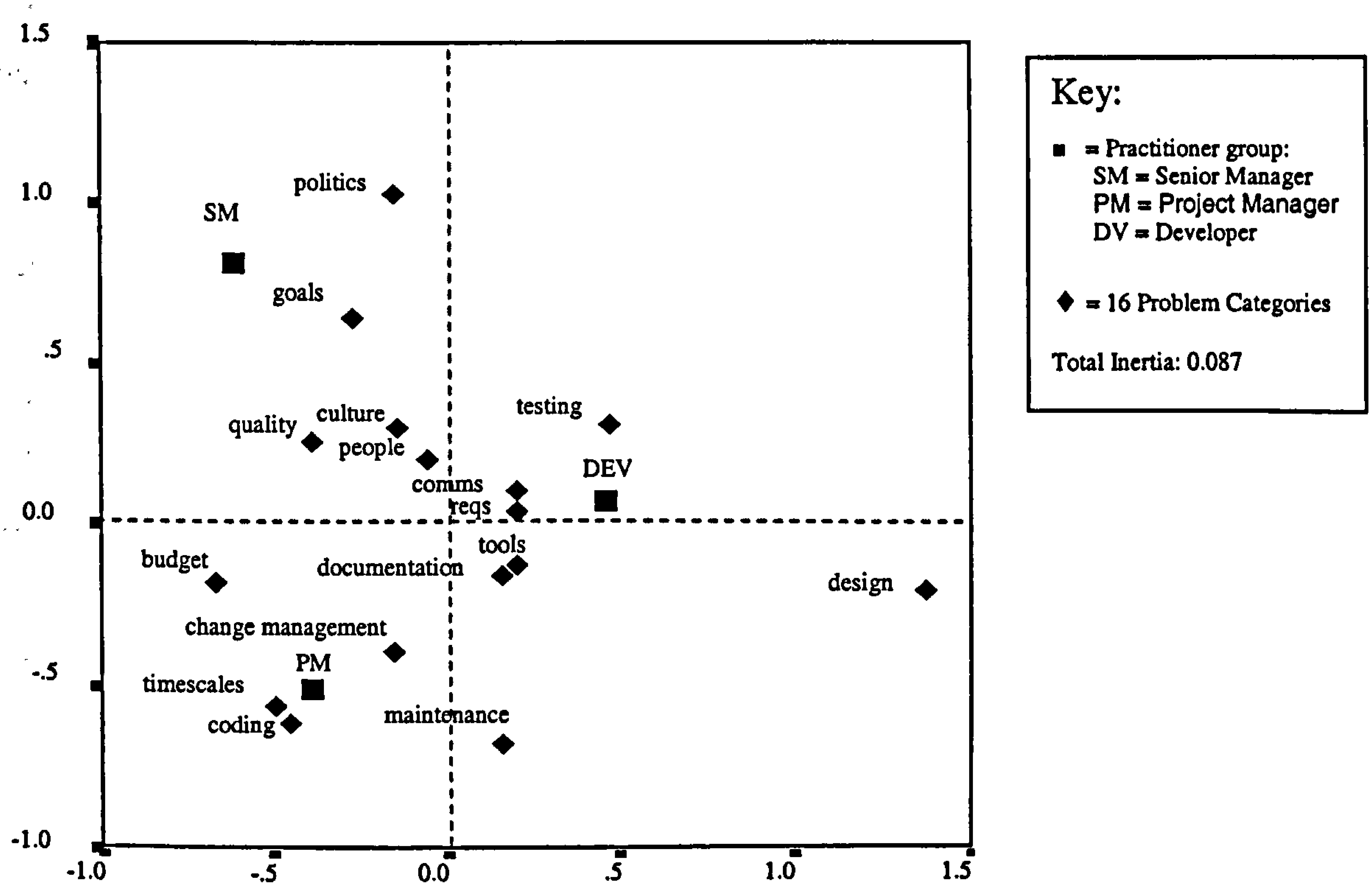


Figure 11 is based on data from row percentages in Table 39, Appendix G. Figure 11 graphically displays problem clusters around staff groups and shows how staff groups inter-relate. The CA map therefore offers a new perspective on how problems and practitioners relate. Using the figures from Table 39 also releases the problem categories from the higher-level classification scheme that could be viewed as too general to be useful (i.e. organisational, project and lifecycle in Figure 10).

The poster in Appendix H summarises the processes involved in identifying, classifying and analysing the focus group data which culminates in a CA description of the data.

4.9.1 Observations drawn from the data

Points of interest are:

- The total inertia of the CA map in Figure 11 is 0.087, which indicates that there is a difference in how each practitioner group view some problems; if all practitioners expressed the same problem concerns the inertia would be 0.00. Extreme difference between groups in every area would result in a high inertia of approximately 0.99.
- The 3 practitioner groups are equidistant from each other and exist in different quadrants, suggesting that each practitioner group is independent in how they cite SPI problems.
- Developers show most concern for RE, Communication, Tools and Technology, Documentation and Testing. Developer's inertia is relatively low (0.022) suggesting that their problems are near to the average.
- Project Managers are concerned with Budgets & Estimates, Timescales and Change Management and Coding as they are all in the same quadrant. Project Manager's inertia is relatively low (0.026) suggesting that their problems are near to the average.
- Senior Managers' inertia is relatively high (0.039), signalling that some of their areas of concern differ from the norm.

- The proximity of Senior Managers to Goals and Politics and their positioning on the periphery of the map suggest problem polarisation.
- People, Documentation, Tools and Technology, Requirements and Communication are closest to the centroid and therefore represent the most common problems.
- There is a distancing from Documentation, Tools and Technology, Requirements and Communication with Developers being the closest, Project Managers further away, and Senior Management furthest away.

The CA map therefore reveals a pattern of association that is lacking in the two dimensional view of problems presented in the problem tables (see for examples, Appendix G). This graphical view places requirements closest to the centre, showing this area to be the most common problem to all groups.

4.10 Discussion of results

This section summarises how the results from investigating the three research questions impact the work in this study.

4.10.1 SW-CMM level problem association

My results relating to research question 1, suggest that there is an association between reported problems and SW-CMM maturity levels. Low maturity companies suffer from project and technical problems while high maturity companies are more burdened with organisational problems.

There appears to be a tension between the advice given by the literature on the one hand, and the guidelines offered by the SW-CMM and my findings on the other hand. The literature states that organisational issues (especially the human element) are of prime importance to the success of SPI initiatives while the SW-CMM focuses on project issues before looking at organisational issues. For example, Moitra suggests that organisational issues are important to successfully introduce, deploy and institute recommended software engineering and management practices (Moitra

1998). My findings show that low level companies are project focussed which indicates a convergence with the SW-CMM. SPI pioneer Humphrey (1989), suggests that management must consider the technical or 'project' issues rather than their organisational needs when embarking on SPI. This is consistent with the SW-CMM, where it is not until SW-CMM level 3 that organisational issues become a KPA. My findings indicate that low maturity companies are not in a position to concern themselves primarily with organisational issues as they tend to have more urgent technical and project problems.

My high maturity company results suggest that such companies have solved most of their low level technical problems and are in a position to focus on organisational problems. Furthermore, my findings also indicate that high maturity companies recognise the importance of people within software development (people issues form a major part of the organisational category). Because many of the people issues reported in my results are outside the scope of the SW-CMM, companies may not be receiving enough support and guidance on the human issues in SPI when following this model exclusively.

Moving from SW-CMM level 1 to SW-CMM level 2 is known to be a difficult advancement. My findings indicate those companies at levels 2 and 3 share similar process problems (also confirmed by swifter movement between these levels). This suggests that there is a similarity in company behaviour at these two levels. Furthermore, it may mean that once a company has moved beyond level 1, it is better prepared for the next process improvement stage.

4.10.2 Practitioner problem group association

The results from investigating research question 2 show that developers, project managers and senior managers report similar problems with their SPI initiatives. They all share a common concern for 'people' issues that, with the exception of 'training', are outside the scope of the SW-CMM. However, practitioners' problem priorities differ to reflect their varied experiences and roles. There is little evidence to suggest that any staff group identifies strongly with another staff group in how they experience SPI problems. Although this difference is to be expected, it could be argued that a problem for any individual staff group is a problem for the company as a whole.

My findings indicate that senior management is isolated from the other two staff groups, with its problems focused around 'goals' and political influences. Although senior managers share a common concern for 'people' issues such as skills shortages, they are not necessarily aware of the issues directly affecting developers and project managers such as documentation and tools and technology. This apparent lack of insight is likely to have a detrimental impact on the ability of senior management to design effective SPI implementation strategies.

4.10.3 RE process problems in context with SPI

The results from research question 3 clearly show the RE process as causing the greatest lifecycle problems to all practitioner groups and SW-CMM levels. The CA map is particularly helpful in highlighting requirements engineering as a common problem to all groups. Although the frequencies in the two dimensional table do not place RE process problems as the most prominent issue, the measure of inertia given in the CA map highlights the prominence of this problem group in relation to all other problems.

As this study is aimed specifically at companies following the SW-CMM methodology, my results suggest that the model may not be supporting practitioners in their RE activities. These findings are consistent with the literature review suggesting that further work is required to investigate how the SW-CMM specifically approaches the RE phase of software development.

4.11 Conclusion

The content analysis approach to data gathering helped to group problems into logical categories. These categories allowed me to explore the relationships between problem and SW-CMM level, and problem and staff group. I have shown the relative importance of these problems and the relevance of the most pressing problems. I have made a distinction between problem groups through clear definitions and comparisons. I have also shown the problems SPI companies are experiencing in their organisational, project and software development life cycle processes. Concrete examples are given of typical problems occurring in software development companies. I developed this theme to highlight areas where problems are concentrated.

Companies in my study are suffering mainly from organisational problems. Within this problem group, there is a concentration of people and communication issues. These problems are common to all SW-CMM levels and all practitioner groups. It is the high-level maturity companies who are most aware of organisational problems, along with senior managers and project managers. This is likely to be because companies with mature processes do not have so many problems at the project level, and developers do not involve themselves with organisational matters. Managers embarking on a SPI effort, therefore, need to be aware of the omnipresent organisational issues, while making sure the project and lifecycle issues are given the appropriate focus, especially the recurring problems developers are having with RE, tools, technology and documentation.

Managers in low maturity companies need to consider lifecycle problems; in particular they should make resources available to manage RE so that companies can progress to the desirable position of the more mature companies where lifecycle issues are no longer a priority problem area. The lack of direct lifecycle development problems observed in the high level maturity companies appears to endorse the SW-CMM by indicating that the higher maturity companies are indeed producing software that is more reliable and predictable. Managers therefore need to conquer the problems associated with lower-level process maturity in order to achieve the benefits associated with the higher-level maturity companies.

Management involved in quality assessments can gain by looking at how each staff group is approaching SPI and they need to recognise that even if the groups share similar company goals their problems are likely to be different. It is therefore important that when creating a model of process improvement individual practitioner needs are considered. If managers are going to achieve a universal 'buy-in' to SPI there must be something in the improvement effort for everyone.

This study gives some insight into how the SW-CMM improvement model is being used in the field. The model, to date, has undergone little empirical testing and it is therefore reassuring to find that the problems companies in the context of this study are experiencing appear to be linked to their SW-CMM level. This suggests that the model is well constructed and as a result managers should have more confidence in using the improvement model and addressing problems voiced by practitioners. Also, as this maturity structure is a possible strength it is worth emulating when developing an augmented model to focus on the RE process.

Chapter Five: Requirements engineering process problems – an empirical study

5.1 Introduction

In chapter four the RE process was identified as a major problem for practitioners in the UK. The literature echoes the findings in chapter four where RE process issues appear to be dominating software development. Categorising the focus group data into major development phases revealed the RE process as accounting for almost half of all development problems. In this chapter I re-visit the data presented in chapter four to produce a finer-grained analysis of the problems inherent in the RE process. My findings provide a fuller understanding of the problems companies experience in their RE processes.

Although the companies in my empirical study varied in size and application area, they were all using the SW-CMM to guide them in their software process improvement activities. A comment from a senior manager shows the wider benefits of implementing a SW-CMM improvement method, "it should help people have a stronger sense of being professional and working for a first class company and should help towards retaining staff and reducing costs". While a project manager takes a more pragmatic view stating that "[the SW-CMM] helps you to control your destiny".

When asked about general problems these practitioners were having with their software development a common theme throughout all focus groups related to requirements engineering. For example a project manager states, "I don't believe that we spend enough time up front of the project doing all the work, understanding exactly what we need to do and consequently we learn as we go through and have to keep changing the requirements". Another quote given by a developer clearly shows a frustration with the lack of control over inevitable changes in requirements, stating: "We get changes in requirements during development which add extra resource factors onto our jobs but that is not taken into account. It is not factored into our time scales. It is the biggest problem for me at the moment". These requirements problems were common throughout the 4 levels of SW-CMM maturity represented in the focus groups.

In this chapter, categorical data drawn from the focus groups are presented in contingency tables, where differences and similarities between groups are observed. The groups of interest are practitioners, finer grained RE process problems and SW–CMM Levels. This study is based on work undertaken with the PPP group as recorded in (Hall *et al.* 2002a; Hall *et al.* 2002b).

5.2 Study aims

This exploratory study aims to answer the following research questions:

5.2.1 Research question 1: Do organisational problems or technical problems impact most on the RE process?

To gain a better understanding of the underlying processes involved in RE, I now look at the finer-grained RE processes as identified in the focus groups. I look at variations in both organisational and technical RE processes to see whether the experiences of practitioners in my study reflect the evidence given in the literature. The findings from this study will inform the RE process model that aims to link best practices with practitioner needs.

5.2.2 Research question 2: Does increased process maturity reduce RE process problems?

I investigate whether high maturity companies have fewer RE process problems than low maturity level companies. Placing problems in context with the SW-CMM will provide a better understanding of how supportive this model is of the RE process. If the SW-CMM maturity structure is not helping practitioners with their RE process it could be argued that it is not worth using the SW-CMM maturity characteristics to form the basis of a RE process model.

5.2.3 Research question 3: Do different staff groups report different RE process problems?

I examine the individual RE process problems of each practitioner group as the success of improvement activities depends on the involvement and buy-in of all those involved in the process (Baddoo 2002). Also, if there is a difference the proposed RE process model should reflect the importance of identifying and involving key groups of RE stakeholders.

5.2.4 Research question summary

Results from the analyses driven by these three research questions will help create a clearer picture of the tensions and priorities within this complex process. Understanding the problems companies are experiencing with their RE process will help to build a picture of where solutions are needed when building a model of the RE process.

5.3 Identifying RE processes

In this chapter I analyse the focus group data as detailed in chapter four. I re-categorise the data in order to abstract finer grained problems that relate to the RE process sub-category of the lifecycle class of problems as well as some of the organisational processes where RE problems were specifically mentioned. The original 'grounded theory' approach that takes a bottom-up view of the data means that these finer-grained categories have already been created. It is in this study that a content analysis approach is taken to the data (Krippendorff 1980) where all the RE process problems are re-organised and clustered into two main groups: organisational-related RE problems, and technical-based requirement problems. The glossary in Appendix A provides definitions of these terms and Appendix J provides additional details of the RE process problem classification scheme derived from the PPP focus group data.

5.3.1 The size of the RE process problem

Table 17 shows how the 45 focus groups in the study report RE process problems. The table shows that 63% of RE process problems can be attributed to organisational factors, rather than to technical factors inherent in the RE process.

Table 17: Classification of RE-related problems

	Frequency	Percentage
*Organisational-based	232	63
*Technical-based	132	37
Total number of development Problems	364	100

(*For definitions of organisational and technical RE problems please see the Glossary in Appendix A)

The details presented in Table 18 suggest that for my sample the organisational issues contributing to RE process problems are quite diverse. However, almost all of the issues cited are human-based. Some of these human issues are internal to the development process, for example those relating to developer skills or staff retention, and some are external to the development process, for example those relating to communication with users. Internal issues that relate to developers appear to be a greater problem for these companies than communication with users. Furthermore, human issues seem to be more of a challenge to these companies than resource issues. For definitions of these RE process problems please see Appendix J.

Table 18: Classification of organisational-based RE problems

	Frequency	Percentage
Developer communication	55	24
Skills and responsibilities	46	20
Resources	34	15
Staff retention	29	13
User communication	30	13
Training	20	9
Company culture	18	8
Total number of organisational Problems	232	100

Table 19 shows over half of the problems emanating from within the RE process are related to poor initial requirements capture, undefined processes and requirements growth. However I am quite surprised that requirements growth (that incorporates requirements change) was not identified as a bigger problem than it was as it is so well documented as a problem in the literature. Furthermore, despite the technical sophistication of some of the products developed by these companies, they seem to have relatively few problems with users understanding of their own needs. A specific communication problem that arose in a few focus groups was related to dissatisfaction with the contribution of marketing and sales departments to the RE process. Indeed one developer commented:

"Customers have got them [the sales department] by the tail now and we can't have that.... we should be saying 'you can't have that, it's not scientifically achievable' but we're not."

Table 19: Classification of technical RE process problems

	Frequency	Percentage
Vague requirements	33	25
Undefined RE process	32	24
Requirements growth	31	23
Complexity of application	27	20
Poor user understanding	5	4
Requirements traceability	4	3
Total number of technical RE process problems	132	100

Looking at the problems in Table 19, it is possible that many of the RE process problems presented are due to the organisational problems presented in Table 18.

5.4 RE process problems and company maturity

Table 20 shows the total number of problems experienced in the RE process relates to company maturity. The figures in the table suggest that the number of problems decreases through the maturity levels. However there is considerable variation within companies at the same maturity level. For example, the six level 1 companies report variable frequencies of RE process problems. This may reflect those companies being at different stages within the level 1 band and illustrate the *ad hoc* nature of level 1 maturity. A company close to attaining level 2 is likely to be different from a company right at the bottom of level 1. It may be that some companies are so immature that they under-report RE process problems, as they are not mature enough to recognise their problems (Finkelstein 1992). Similarly the level 3 companies report quite a high level of RE process problems. These companies are probably mature enough to identify many of their RE process problems and to be actively seeking out process weaknesses. Level 3 companies are also mature enough to ensure everyone is aware of problems, but not mature enough to have solved all their problems. This may explain the high number of problems reported by Company 13.

Table 20. Maturity and RE process problem frequency

Company	CMM Level	Freq'cy Org'l Probs	Freq'cy Req Procs	Total Req Probs	%	CMM level mean	CMM level %
1	1*	30	18	48	13%		
2	1	23	3	26	7%		
3	1	22	5	27	7%		
4	1	41	26	65	18%		
7	1	8	6	13	4%		
10	1	16	15	31	9%	35	33%
8	2	22	9	31	9%		
11	2	10	18	28	8%	30	28%
6	3*	17	9	26	7%		
9	3	5	8	13	4%		
13	3	24	14	38	10%	26	24%
5	4*	15	1	16	4%	16	15%
TOTALS		232	132	364	100%	106	100%

* indicates formal SW-CMM assessment

The difference in sample sizes prevents making a direct comparison between how each SW-CMM level is reporting RE-related problems. "By itself the score will be meaningless ... unless we can place such a score in context" (Oppenheim 2001). The frequencies have therefore been normalised in each category. This normalisation is shown in Table 21.

Table 21: SW-CMM level problems

	CMM Level 1 (6 companies)		CMM Level 2 (2 companies)		CMM Level 3 (3 companies)		CMM Level 4 (1 company)		Total	
	Obser'd	Norm'd (mean)	Obser'd	Norm'd (mean)	observ	Norm'd (mean)	Obser'd	Norm'd (mean)	Obser'd	Norm'd
Organisational problems	139	23.17	32	16	46	15.33	15	15	232	417
Technical problems	73	12.17	27	13.5	31	10.3	1	1	132	222
Total	212	35.34	59	29.5	77	25.63	16	16	364	639

The normalised figures from Table 21 are presented in Figure 12 in order to explore the data further. The figure reveals that although there is an improvement in technical problems, the organisational problems in SW-CMM levels 2-4 are untouched by the improvement effort.

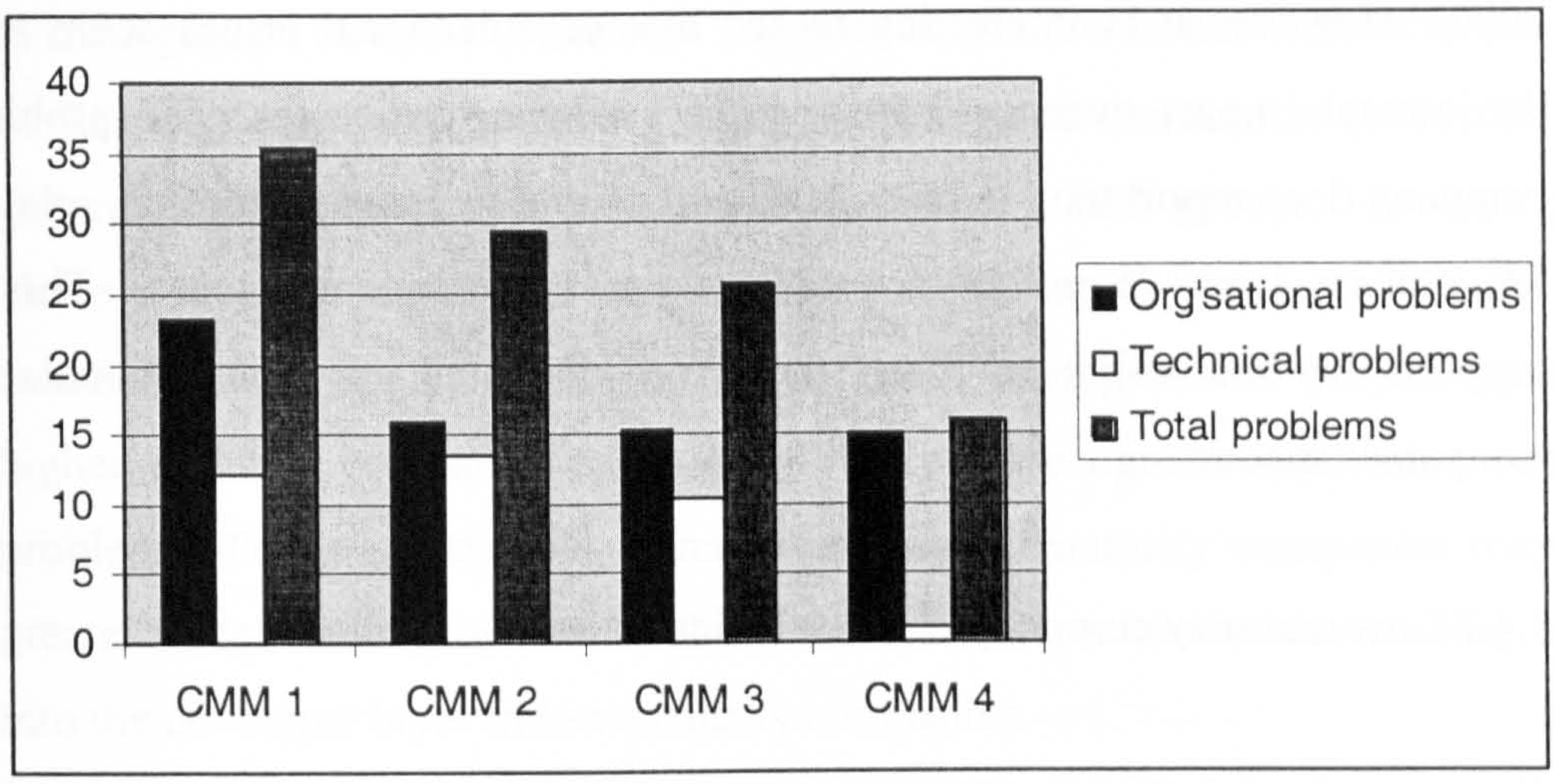


Figure 12: RE problems by SW-CMM level (using normalised data from table 21)

A chi-squared test ($X^2 = 9.38$, df = 3, p = 0.02) indicates that there is a significant association between SW-CMM maturity and problem types. However, as the data has been manipulated, Table 21 gives both the observed and normalised figures. The differences highlighted in the bar chart suggest that separating organisational from technical RE process problems is worthwhile. Viewing the RE process as a whole (as shown by the grey bar in Figure 12), would, in this case, mask the weakness in the organisational class of problems.

5.4.1 High maturity company characteristics

Company 5 has been formally assessed as having a level 4 software process capability. As there is only one company in the sample representing this high level of maturity, it is not appropriate to generalise from these results. However I make the following observations about this company.

Tables 40 and 41 in Appendix K list company maturity levels separately. Figures in these tables show that overall the high maturity company in my study, Company 5, experiences relatively few RE process problems with only six cited (excluding the culture hotspot). This is re-assuring as it suggests that Company 5 actually has implemented an improved RE process. This is consistent with El Emam and Birk's findings that higher software RE process capability is associated with better project performance in their validation of the ISO/IEC 15504 model (El Emam and Birk 2000). However it is interesting to see that organisational issues seem to have a detrimental impact even on high maturity software processes. The problems this company does report tend to be cultural and people-oriented rather than related to the RE process itself (5 out of 6 problems are organisational issues). This finding supports the increasing emphasis the SW-CMM puts on organisational issues as companies mature.

5.4.2 Low maturity company characteristics

The strength of feeling in Company 1 regarding 'undefined RE processes' is shown in Table 43 in Appendix K. This issue accounted for 50% of this company's technical RE problems which was higher than any other requirement problems cited by the remaining five level 1 companies. I suspect that this is related to Company 1 having been formally SW-CMM assessed at level 1. It is likely that the assessment process has made everyone very aware of RE process deficiencies. RE process 'hotspots' occur in other companies, for example:

- Companies 3 and 5 focus on requirements growth
- Companies 7 and 10 focus on vague initial requirements
- Company 8 focuses on the complexity of the application

5.5 Staff groups and problem patterns

I analysed the data according to the problems reported in developer, project manager and senior manager focus groups. Table 22 shows that developers generally report more RE-related problems than the other two staff groups. Although this is partially explained by the increased number of developer focus groups conducted, even when the data is normalised to account for this, developers report many more problems than either manager group. This suggests that developers are more aware of inadequacies in the RE process than managers are. On the other hand it could be that managers are not as forthcoming about problems as developers – a phenomenon noted in other work, for example Hall and Fenton (1996).

A finer grained analysis of the data suggests only a few issues with different patterns of RE problems between staff groups. Herbsleb and Goldenson (1996) also report general similarities in data collected from different staff groups. A result given in Table 44 in Appendix K, suggests some differences between the staff groups associated with application complexity. Senior managers and project managers in higher maturity companies seem more likely to recognise application complexity problems. This suggests that managers in higher maturity companies may have a greater understanding of low-level RE issues, whereas that understanding is locked into the developer layer in low maturity companies.

Table 22: RE process problems across staff groups

	Co.1 CMM Level 1*		Co.2 CMM Level 1*		Co.3 CMM Level 1+		Co4 CMM Level 1+		Co.5 CMM Level 4*		Co.6 CMM Level 3*		Co.7 CMM Level 1+		Co.8 CMM Level 2+		Co.9 CMM Level 3+		Co. 10 CMM Level 1+		Co 11 CMM Level 2+		Co 13 CMM Level 3+		Total Row Frq	Total Row %
	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%		
Developers total	24	50	14	56	6	22	24	36	5	31	21	81	3	21	14	45	5	38	21	68	16	57	36	95	189	52
Req process probs	*10*		*1*		*4*		*11*		*0*		*6*		*3*		*3*		*3*		*10*		*10*		*14*			
Org'l req probs	*14*		*13*		2		*13*		*5*		*15*		*0*		*11*		2		*11*		*6*		22			
Project managers total	19	40	5	20	21	78	35	52	8	50	5	19	4	29	8	26	0	0	8	26	12	43	2	5	127	35
Req process probs	*7*		2		*1*		*12*		*1*		*3*		*0*		*1*		*0*		*5*		*8*		*0*			
Org'l req probs	*12*		*3*		*20*		*23*		*7*		2		*4*		*7*		*0*		*3*		*4*		2			
Senior managers total	5	10	6	24	0	0	8	12	3	19	0	0	7	50	9	29	8	62	2	6	0	0	0	0	48	13
Req process probs	*1*		*0*		*0*		*3*		*0*		*0*		*3*		*5*		*5*		*0*		*0*		*0*			
Org'l req probs	*4*		*6*		*0*		*5*		*3*		*0*		*4*		*4*		*3*		2		*0*		*0*			
Column Total	48	100	25	100	27	100	66	100	16	100	26	100	13	100	31	100	13	100	31	100	28	100	38	100	364	100

* indicates formal SW-CMM assessment + indicates SW-CMM level is based on self-assessment

In addressing the first research question:

Research question 1: What pattern of RE process problems are companies experiencing?

Most of the RE process problems experienced in the companies in my study were organisational. Viewing RE processes as a whole would mask the difficulties practitioners are experiencing with the organisational processes that support the technical RE process. Furthermore my findings suggest that organisational issues exacerbate all types of RE-related problems. For example, lack of skills and poor staff retention seem to have a significant impact on the capability of the RE processes to produce good initial sets of requirements.

Problems inherent in the RE process itself did not seem to be presenting major difficulties to companies. When placing these internal problems in context with organisational problems even requirements growth does not feature as a major problem. However, the relatively low number of RE process problems reported that relate to complex and highly technical issues may be due to the general nature of the discussion. A further reason may be due to developers deflecting blame, as most of the practitioners involved in the focus groups were highly involved in development processes. This may have created an incentive for participants to cite problems stemming from outside development processes. Also, the low number of references to requirements traceability problems could be due to the traceability process being viewed as a solution, and a 'lack' of traceability may not be easy to identify.

In addressing my second research question:

Research question 2: Does increased process maturity reduce RE process problems?

My results suggest a relationship between RE process problems and process maturity. There seem to be RE benefits available to high maturity companies. Indeed process assessment seems to generate benefit to companies even when they are assessed at level 1. Although Company 1 is the only company in my sample to have been formally assessed at level 1, the strength of feeling this company exhibits regarding its poor RE process is interesting. I suspect that the formal assessment process has made people in Company 1 acutely aware of the deficiencies in their RE process. This ties in with Herbsleb and Goldenson's findings that 90% of SW-CMM assessed companies in their study experienced particular improvements as a direct result of the assessment exercise (Herbsleb and Goldenson 1996).

The other level 1 companies who have not had a formal process assessment may not yet be aware of all their own problems. Yet, improved understanding of problems is the only way in which improvement can occur. My results also show that staff retention is a problem for many of the companies in my study. However, it seemed more of a problem to the lower maturity companies than to the higher maturity companies. Again, my findings correspond with those in published case studies. Reports describing Siemens' progression from low maturity say that one of their problems was high staff turnover (Paulish and Carleton 1994). Similarly reports from Schlumberger identify problems retaining SPI staff (Wohlwend and Rosenbaums 1994). Whereas Boeing, a level 5 company, found staff retention less of an issue (Yamamura 1999). Similarly, at Oklahoma City Air Logistics Center the same people remained on the Management Steering Team for 10 years (Butler and Lipke 2000). The suggestion is that high maturity companies retain staff, not that high maturity companies can necessarily sustain a high turnover of staff.

In addressing my final research question:

Research question 3: Do different staff groups report different RE process problems?

My results suggest that there are some differences in the problems reported by developers and manager groups. Developers seem to show a better understanding of RE process problems than manager groups. Furthermore, developers report more RE process problems than either manager group. This generally supports the view that developers should be involved in the design and improvement of RE processes. Bach (1995) along with others are strong advocates of involving developers in process improvement. Furthermore, NASA's Space Shuttle Project reports that involving developers in process work played a critical role in achieving its level 5 status (Billings and Cliffton 1994).

Communication between developers and users via sales and marketing staff was a deeply felt problem in a couple of the companies in my study. Development staff felt that sales staff agreed to deliver unrealistic system features without considering technical and schedule implications. Such conflict has also been identified in other work (Hofmann and Lehner 2001; Yu and Mylopoulos 1997).

My finer grained study of RE process problems also revealed that developers in the companies rarely speak to customers or users. Indeed a developer told us that an informal chat with a user at a company Christmas party achieved more than months of formal requirements capture. Reports from high maturity companies show that improving communication between developers and customers is an important area. For example, the space shuttle project achieved a 75% decline in RE process problems "in part because the customer became more aware of requirements issues" (Paulk 1993).

My results suggest that developers are generally unconcerned about users not understanding system needs. Although poor initial requirements were considered problematic, developers did not seem to blame users for this. Sales and marketing were often considered the culprits. This may indicate that user understanding of their own needs is improving, though there is little evidence to show this and commentators continue to report that users often do not know what they need (Potts 1993). On the other hand, it may be that developers are altering their perceptions of the roles played in establishing good requirements – developers may be shifting responsibility away from users as recommended by Scharer (1990).

5.6 Conclusion

Many commentators speculate on problems in the RE process. The results in this study contribute empirical evidence towards the scale and shape of RE process problems. Although it is not possible to generalise from my results, they do offer insights into the RE process problems of the twelve software companies in this study.

Predictably developers are shown to be more aware of inadequacies in the RE process than project managers and senior managers. However, the analysis highlights that the areas giving developers the most concern are human-related and of an organisational nature. Organisational issues appear to be more important than technical problems in RE processes. They do not seem to be addressed in a planned way and are likely to amplify some RE process problems.

The SW-CMM appears to be helping companies with their RE process problems to an extent. Higher maturity companies tend to exhibit fewer RE process problems. Those RE process problems exhibited in higher maturity companies tend to stem from organisational issues rather than development process issues. High maturity requirement processes seem to be more resistant to 'damage' from organisational issues. Manager groups in high maturity companies exhibit a greater understanding of requirement process problems than manager groups in low maturity companies. Best practices can be learnt from higher level SW-CMM companies, as they exhibit fewer problems with RE processes.

This study has clarified the needs of practitioners and therefore guides and influences the design of the model I propose to develop to support the RE process. For example the model should include a process to consult and involve key stakeholders to include developers as they are most aware of RE process inadequacies. Also, to improve the RE process it is critical to consider methods for improving communication between developers and customers.

The findings from this study further suggest that the proposed RE process model should include and integrate organisational processes with the technical processes. These human-based problems are occurring in all levels of process maturity as characterised by the SW-CMM. Also, the model should include an assessment component to help organisations prioritise their RE process problems. This is because each organisation is likely to have unique priorities governed by their own particular company goals.

Overall the results given in this chapter contribute to the increasing body of knowledge showing that improved RE capability is related to improved organisational performance (El Emam and Birk 2000). As noted by Frangos (1998), the capability of technical processes will be liberated only when non-technical issues are improved. Furthermore my findings indicate that immature companies are especially susceptible to problems in the RE process. Given that 70% of software companies are said to remain at SW-CMM level 1, the scale of RE process problems across the industry could be very large. Considering the criticality of RE to project success, it appears that software companies are in need of further support with their RE process. In the next chapter, therefore, I consider how to integrate the findings in this chapter into a model that views RE processes within the SW-CMM framework.

PART THREE

REQUIREMENTS ENGINEERING SOLUTIONS

Chapter Six: Building a model of the RE Process – a top down approach

"Many of us in the software engineering field have long believed that systematic, rigorous engineering approaches to software development must start with systematic, rigorous approaches to requirements engineering. Finding ways to apply these approaches in practice on a wide scale has been the stumbling block" (Cheng and Weiss 2000 page 20).

6.1 Introduction

In this chapter I describe my approach to building a maturity model to reflect best practices in the RE process. This model is based on the problems highlighted in the previous two chapters. At this initial stage of development I present a top-down view of the R-CMM to show how the model is moulded by SW-CMM architecture and concepts (Paulk et al. 1995). Key RE processes are identified, defined and prioritised according to the prescribed maturity structure. This initial work on model development is covered in (Beecham *et al.* 2003b; Beecham *et al.* 2003c).

The primary motivation for building the Requirements Capability Maturity Model (R-CMM) emanates from my empirical research with 12 software development companies as described in chapter four. My research highlighted problem areas in software development that led to a detailed study of the problems practitioners were experiencing in their requirements engineering activities as presented in chapter five. My studies examined the first four SW-CMM levels. A primary aim of the RE process model is to help organisations agree on a strategy for improvement and achieve a consensus on how to implement requirement related improvement activities.

Although the literature provides improvement guidelines and models it is usually difficult to uncover the model development process used. Furthermore, there is very little in the literature to guide model building. In this chapter I outline the activities involved in creating the model to reveal the model's underlying characteristics and show how it might successfully be employed in practice. This transparency will allow for study replication, will add to the validity of the model and will assist users to tailor and implement their own improvement model.

This chapter is organised as follows: Section 6.2 presents a rationale for building the SW-CMM based on previous work. Section 6.3 gives an overview of the modelling development process. Section 6.4 defines the R-CMM and details my objectives for building the model. Seven criteria are identified that create the foundation for model building activities. Section 6.5 outlines how SW-CMM characteristics are converted into a specialised model of the RE process. This section includes a high level view of the R-CMM derived from a SW-CMM framework, and a dynamic view to the R-CMM that includes information flows. It is in section 6.6 that each of the five levels of the R-CMM is viewed in more detail. This presentation includes relevant empirical findings together with SW-CMM characteristics. Finally, section 6.7 summarises and concludes the issues covered in this chapter.

6.2 Rationale for building a model based on the SW-CMM

6.2.1 A problem is identified

My empirical research led me to conclude that the SW-CMM in its current form is not helping practitioners to:

a) identify both technical and organisational RE processes
b) define both technical and organisational RE processes
c) recognise RE process problems
d) assess and agree requirement improvement priorities
e) relate RE process problems to requirement improvement goals
f) relate requirement improvement goals to general SW-CMM guidelines and activities

The literature highlights many limitations in the SW-CMM, suggesting improvements can be made to the structure and content. More specifically, my empirical research points to weaknesses in SW-CMM support of the RE process. It appears that RE needs are not always identified and included in company improvement goals.

6.2.2 A solution is proposed

Many companies throughout the world use the SW-CMM as their software process improvement model. Case studies have shown that the use of this methodology is generally positive with improved processes leading to higher quality software. Even though this evidence may not reflect the state of the software community as a whole (Baumert 1994; Fox and Frakes 1997), SEI records show increasing numbers of organisations follow SW-CMM guidelines (SEI 2003b).

Furthermore, software engineering experts believe that creating solutions that are based on previous work and frameworks will help to progress software improvement (Humphrey 2002). According to Weigers (1998b), once the practical limits of known approaches have been reached, we can turn to improved models that provide guidance for working in better ways. Therefore, as the current SW-CMM approach to improvement seems to be 'necessary but not sufficient ... and does not address many crucial processes or areas of activity' (Rogoway 1998), I create an augmented, specialised SW-CMM to fill this gap.

The specialised RE process improvement model aims to isolate the RE process and assist practitioners to identify and prioritise their problems. Taking the advice given by Paulk et al (1995) the R-CMM guides practitioners to focus on "a limited set of activities" and "work aggressively to achieve their goals".

6.3 Model development process

Figure 13 outlines the stages involved in creating the R-CMM. The first stage in model development is to set the criteria for success using a rule-based framework. These rules govern and guide subsequent model building activities. Stages 1 and 2.2 are covered in this chapter. Stage 2.1 involves an analysis of the data presented in my empirical studies in chapters four and five, and stage 2.3 draws on best practices in the RE literature as presented in chapter two. The following chapter brings the three sources together in a detailed example of the R-CMM. The final stage is covered in chapter eight where an evaluation of how well the model meets the criteria outlined in this chapter. Although stage 4 is the final stage in this thesis and represents the first cycle of model development, the feedback gained from the evaluation will be used to inform future model development.

Figure 13 shows the cycle of development in the five studies:

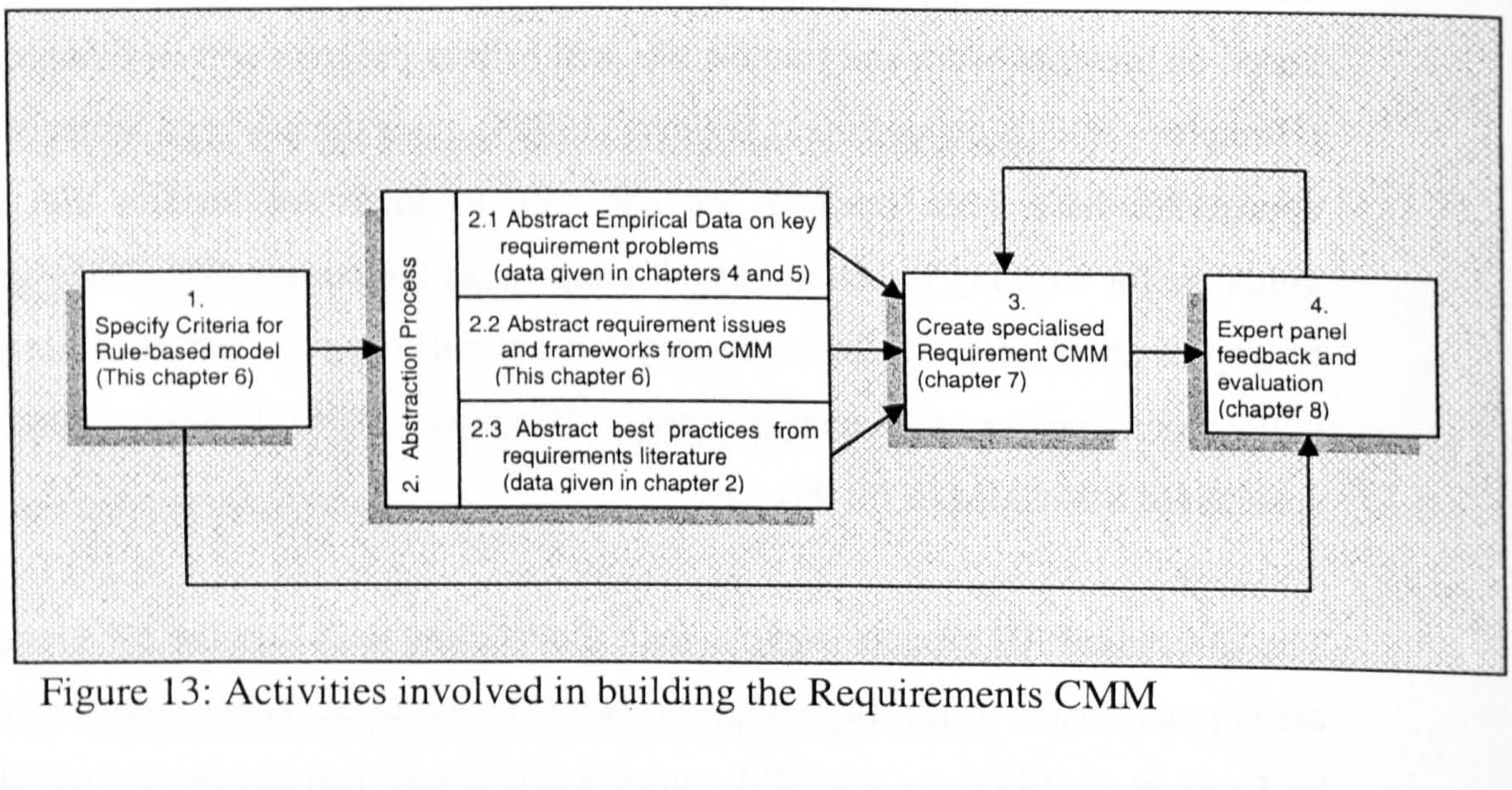


Figure 13: Activities involved in building the Requirements CMM

6.3.1 A modelling framework

Modelling comprises a complex series of activities and it is therefore helpful to gain another perspective on this process. Srinivasan and Te'eni's (1995) empirical study has a similar focus to my own as they examine the activities involved in modelling. Table 23 uses their dynamic view of modelling data as a framework for explaining the activities performed when building the R-CMM. There is some overlap to my presentation, as can be seen by looking at how stages 1 and 2 in Figure 13 map to stage 1 in Table 23. I build on Srinivasan and Te'eni's generic approach to model building shown in Table 23, by adding precise model building activities and their purpose at each modelling stage. Adding this cognitive dimension to my original methodology will help researchers and practitioners gain a further understanding of the rules and strategies that underpin model development.

Table 23: Five cognitive stages of model building

Modelling stages	(Srinivasan and Te'eni 1995) modelling process	The R- CMM modelling process
1	Seek information about goals, objects and actions from external sources	I create rules and criteria for building the model. Goals are abstracted from the SW-CMM, behaviour from my empirical study, actions from best practices in the literature
2	Translate the information from external sources in the light of previous knowledge	The SW-CMM is used to frame knowledge abstracted from the sources outlined in the 1st stage of model building into a logical order and structured format.
3	Internally represent objects, relationships, actions and strategies	A Goal Question Metric (GQM) approach is adapted to include processes to represent the objects, their relationships and actions. Strategies are given in the guidelines
4	Use tools to externally represent the objects, relationships and actions	An assessment procedure guides users through a series of queries to gain an external view of how each process is approached, deployed and how effective it is in practice
5	Test and as a result, refine internal & external parts of problem representation	A prototype model is tested through an expert panel. Internal and external parts of the model are refined as a result of the feedback gained from this study

6.4 Defining the model

The R-CMM mirrors the five level maturity structure of the SW-CMM. SW-CMM RE related processes are integrated with solutions from the RE literature. The model I develop is defined as follows,

The Requirements Capability Maturity Model (R-CMM) is an external and explicit representation of a part of software development that is designed to help practitioners to understand, to change, to manage, and to control the RE process through prioritised best practices within a recognised framework.

The R-CMM is primarily a tool for assessing the current status of the RE process and is an aid to thinking about how best to apply recommended practices. The model supports decision making by regulating the order in which processes are implemented, as dictated by the maturity structure.

6.4.1 Model objectives

Model development is initiated by considering the objectives of the study. Criteria are set to clarify the purpose of the model and to outline what the model is expected to describe. Having a clear set of objectives will help to steer model development and creates criteria against which the model can be tested for correctness and completeness. This process formalises the model and sets out rules to create a firm foundation and provides a structure for the building process. The rule-based development technique is particularly relevant to the modelling of processes. Criteria given in Table 24 initiated R-CMM development and created a working framework.

Table 24: Criteria for R-CMM development

Criteria	Purpose	Rule
Adherence to CMM characteristics	The new model should be recognisable as a derivative of the SW-CMM– both in structure and concept By tapping into the SW-CMM the requirements model takes the strengths of a proven improvement structure and becomes more accessible and compatible, avoiding redundant activities.	- CMM maturity model levels must be implemented - Each level should have a theme consistent with CMM - Key Requirements processes must be integrated - The model should be recognisable as a SW-CMM offshoot
Limited Scope	The model endeavours to be a simplification of the complex system it represents and therefore does not include all RE processes. Sub processes are included on a priority basis as highlighted in the empirical study (Hall et al, 2003). Goals, requirements phases and RE processes define the boundaries of the model.	- Activities relating to technical and organisational RE processes will be included - Processes will be included on a priority basis. - Only processes directly relevant to the R-CMM process areas will be included - Processes will be generic and abstract to allow for individual adaptation
Consistency	R-CMM features need to be consistent and complete at this level of development. Having an acceptable level of 'construct' validity will help users navigate within levels of maturity as well as between different levels of process maturity. Model development and adaptation depends on an acceptable level of consistency.	- Language will be consistent with SW-CMM - Language between and within maturity levels will be consistent. - Structure between model components at similar level of maturity (depth) and different levels of maturity (breadth) will have a consistent granularity.
Understandable	All users of the model should have a shared understanding of the RE process in order to identify where improvement is needed. There should be no ambiguity in interpretation, especially when goals are set for improvement.	- All terms should be clearly defined (i.e. have only one meaning). - All relationships between processes and model architecture should be unambiguous and functional.

Ease of use	Over-complex models are unlikely to be adopted as they require extra resources, and are often too challenging for the user to interpret and follow without extensive training. The model will have differing levels of decomposition starting with the most high level in order to gradually lead the user through from a descriptive model towards a more prescriptive solution	- The model should be decomposed to a level that is simple to understand - Simplicity should be balanced with meaning - The chunks of information should clearly relate as they develop into more complex structures - The model should require little or no training to be used
Tailorable	The model must be structured so that it can be extended and tailored to particular development environments	- The structure must be flexible - The structure must be modular - The structure must be transparent
Verifiable	To assess model strengths and weaknesses the criteria need to be verifiable. Validation of the model will help to improve the model, add confidence in its representation and help with research in this area.	- The objectives set at the outset of the model development must all be verifiable, i.e. I must be able to ask whether my model has met the objectives set out in this table. - Seek external validation

6.5 Converting the SW-CMM

I am aware that the SW-CMM is not a perfect model of SPI. A growing body of literature highlights some of the model's limitations. Fundamental design flaws include weak links between process improvement goals and customer expectations, contradictory sets of assumptions about organisational culture and order of process implementation; vague and incomplete sets of processes, e.g. (Brodman and Johnson 1994; Hall *et al.* 2002a; Hayes and Zubrow 1995; Lauesen and Vinter 2001; Ngwenyama and Neilsen 2003; Sommerville and Sawyer 1997).

Despite recognised weaknesses, as noted above and in chapter two, there are many compelling reasons for using the SW-CMM as a basis for creating a specialised RE process improvement model:

- Pragmatism (it is the most used software process improvement model)
- Tailorability (it is a normative model designed to be adapted)
- Support (it is a 'living' model that is continually being updated by the SEI)
- Empiricism (my original motivating data emanates from companies who use SW-CMM)
- Results (benefits reported include decrease in costs and development time, increase in productivity and quality, for example, see (El Emam and Birk 2000)).

Emulating existing modelling strategies, e.g. (Abdel-Hamid and Madnick 1989; Dybå 2000), my empirical data, together with the literature populate the R-CMM. I have first stripped the SW-CMM to its bare structure whilst retaining specific detail relating to the RE process. In line with the SW-CMM (Paulk *et al.* 1995), the R-CMM aims to define processes at incremental levels of maturity. Maturity levels are characterised by sets of RE processes that are key to software development. One of the aims of the R-CMM, therefore, is to highlight RE practices that appear buried in the all-encompassing SW-CMM.

In agreement with the SW-CMM, for example, a company with an immature RE process is likely to have very few standards in place and could be viewed as having an ad-hoc RE process. On the other hand, a company with a mature RE process will follow a set standard that produces a predictable and stable output. Having a reliable RE process will help organisations to build software that meets customer's needs, is realistic in terms of predicting price and allocating resources and time. Reaching the optimising top level of maturity suggests that the RE process can cope with changes and enhancements with minimal disruption. The R-CMM is designed to work with the SW-CMM improvement programme to evaluate, understand and identify potential weaknesses in the existing RE process. To evaluate these strengths and weaknesses the model includes a method for 'assessing' RE process maturity levels.

Figure 14 shows the three stages involved in adapting the SW-CMM framework to a specialised RE process model. In stage 1, SW-CMM characteristics are converted to form R-CMM level goals, then, in stage 2, the underlying structure is examined to understand the SW-CMM inputs and outputs. Lastly I analyse the content of the SW-CMM in order to extract best practices that are relevant to the RE process.

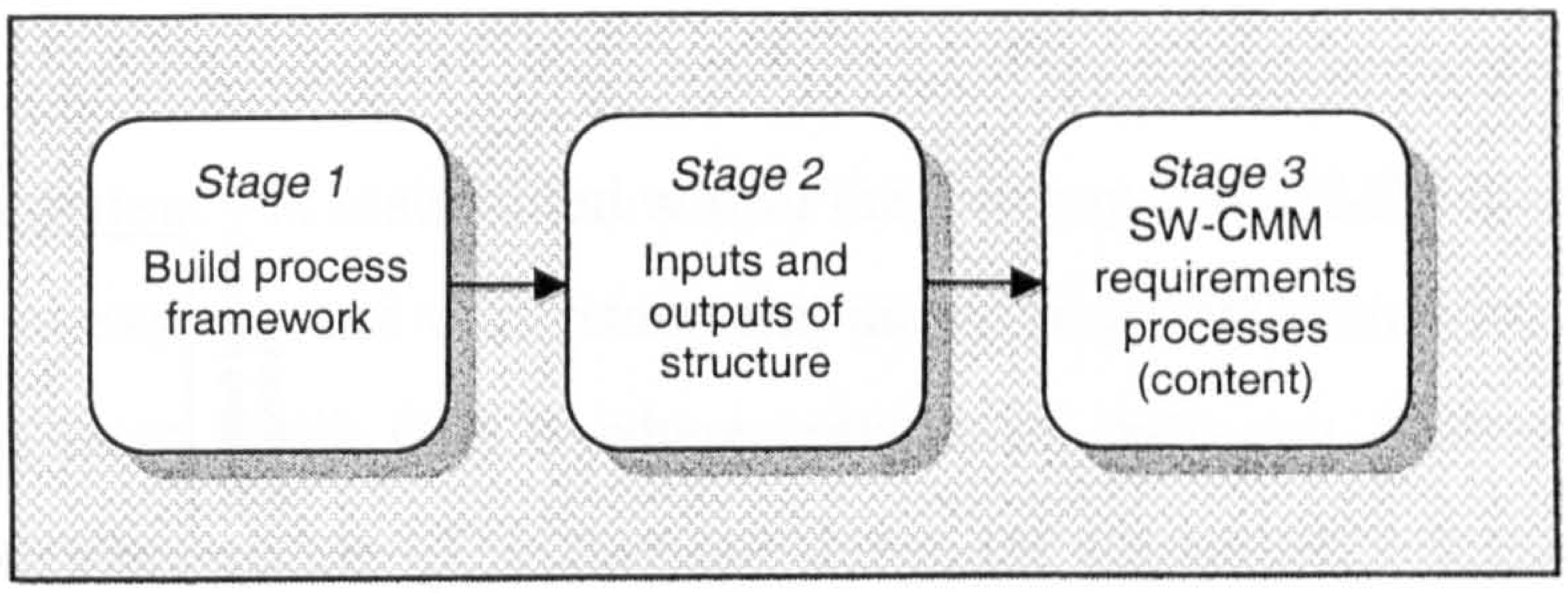


Figure 14: Adapting the SW-CMM to create a framework for the R-CMM

The practices covered in the SW-CMM are not all relevant to the specialised RE process model. Consequently I start with a high level 'static' view where I retain only the maturity concepts from SW-CMM. Figure 15 shows how the R-CMM retains the five levels together with the maturity characteristics that are used to create high-level RE process goals. This initial model is a simplification of a complex system that I continue to develop. I endeavour to capture the purpose of the SW-CMM that is to describe good [requirements] management and [requirements] engineering practices as structured by the maturity framework (Paulk *et al*, 1995).

6.5.1 A top-down view of the R-CMM

The R-CMM is designed to help practitioners strengthen their RE process by implementing practices in a logical order. Figure 15 introduces the R-CMM and places it in context with the SW-CMM. This high level view of the model shows how the RE process matures from an ad-hoc undefined level to a continuously improving level. The model also shows how each R-CMM level has a pre-defined goal to help companies focus on their improvement activities.

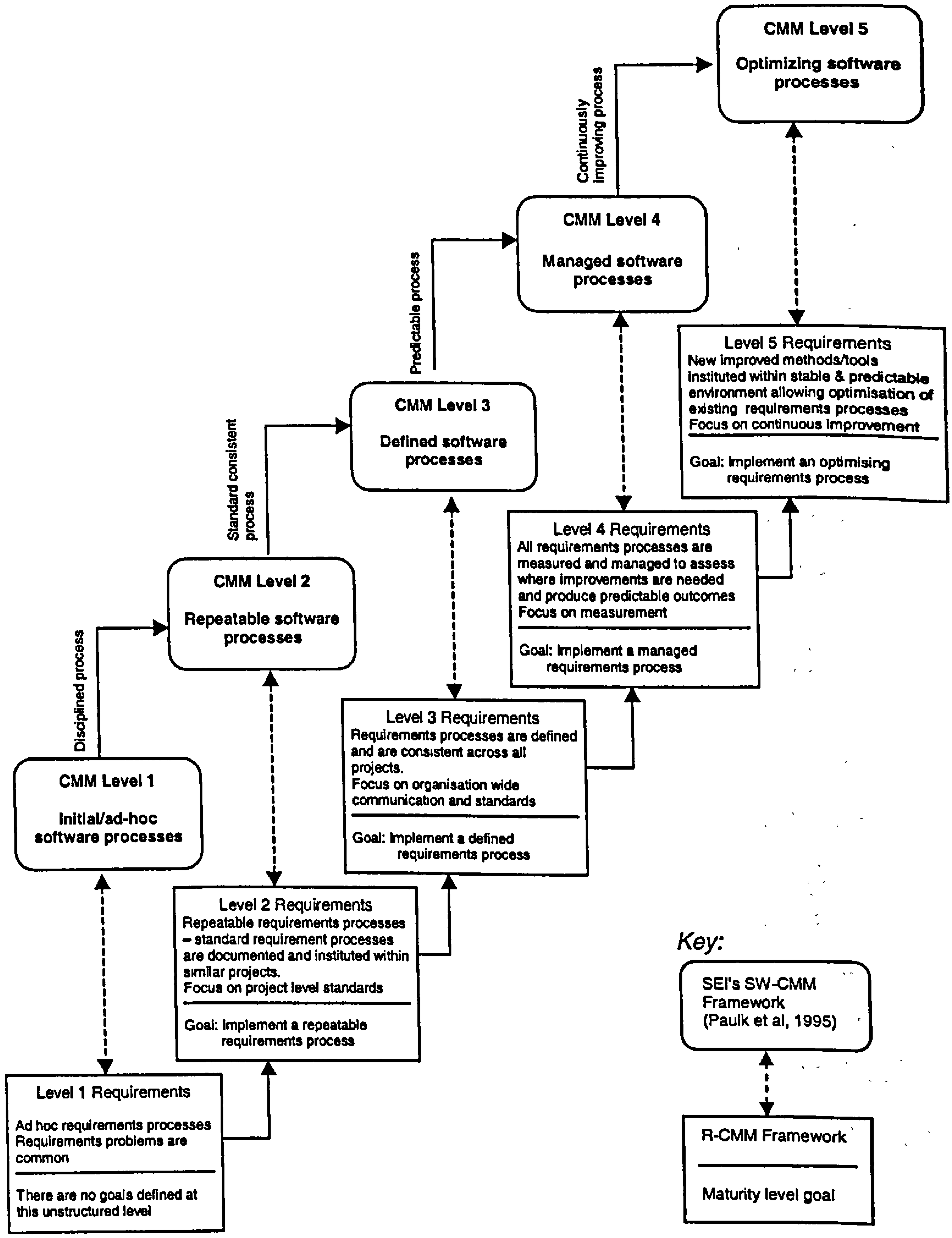


Figure 15: The R-CMM 5 level framework

6.5.2 The dynamic structure of the R-CMM

To ensure consistency is maintained within the existing SW-CMM maturity levels, it is necessary to explore and understand the underlying 'dynamic' structure. Figure 16 is adapted from Paulk (1997) where constituent parts are decomposed and RE process features are added. The diagram shows the relationship between processes, maturity levels and their required inputs and expected outputs. It demonstrates how RE maturity levels indicate process capability and how SW-CMM concepts and empirical findings feed into these maturity levels. Each maturity level (with the exception of level 1) is made up of key RE processes. And, when in place, these key RE processes address clearly defined goals. The model is generic to allow for wide applicability and tailoring to individual company needs.

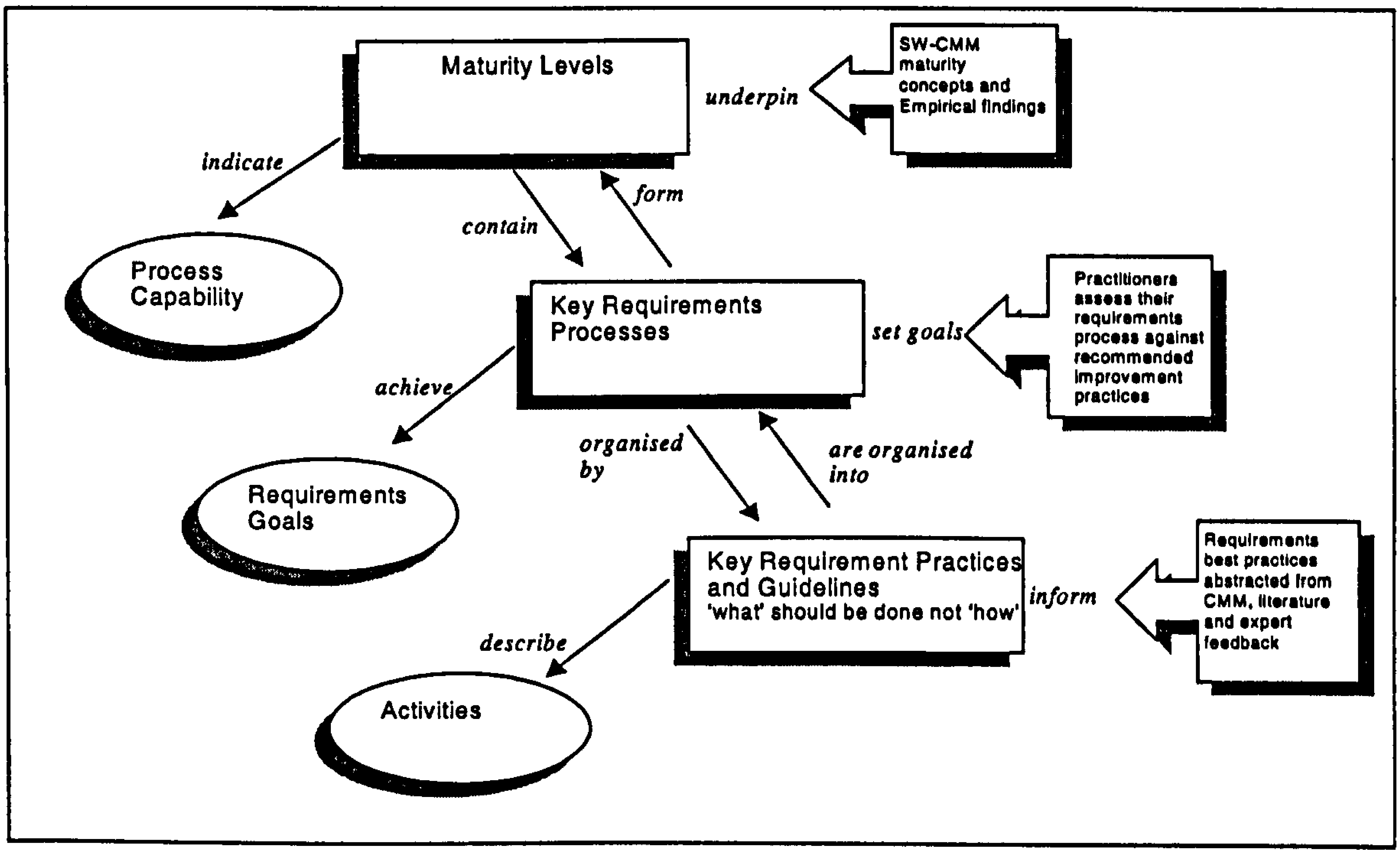


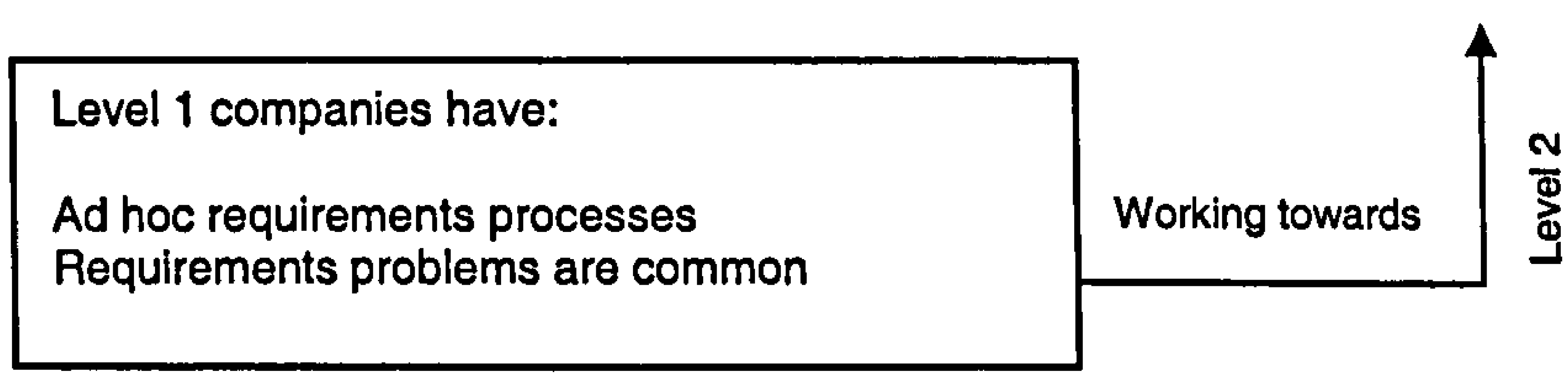
Figure 16: The R-CMM structure adapted from Paulk et al, 1995: p.31

Figures 15 and 16 show how the SW-CMM maturity concept is retained and maturity level goals are introduced. The general goals in Figure 15 are provided by the SW-CMM and are determined by the CMM level characteristics, i.e. the Level 2 goal is to implement a 'repeatable' RE process. In the Figure 16 representation, the goals become more defined as users are guided towards identifying their own needs and relating them to set maturity goals through interacting with the model and interpreting the set RE processes.

6.6 Transposing SW-CMM characteristics into a R-CMM

This section presents an overview of how each of the five levels of maturity introduced in Figure 15 characterises a different RE process capability. Each of the sub-sections below detail how SW-CMM maturity level characteristics act as initial requirement improvement goals. The SW-CMM areas under analysis are: requirements definitions, requirements goals, requirements commitment to perform, requirements ability to perform, requirements activities performed, requirements measurement and analysis; and verifying requirements process implementation.

6.6.1 R-CMM Level 1



(Figure 15 Segment)

There are no process improvement goals defined at this unstructured level.

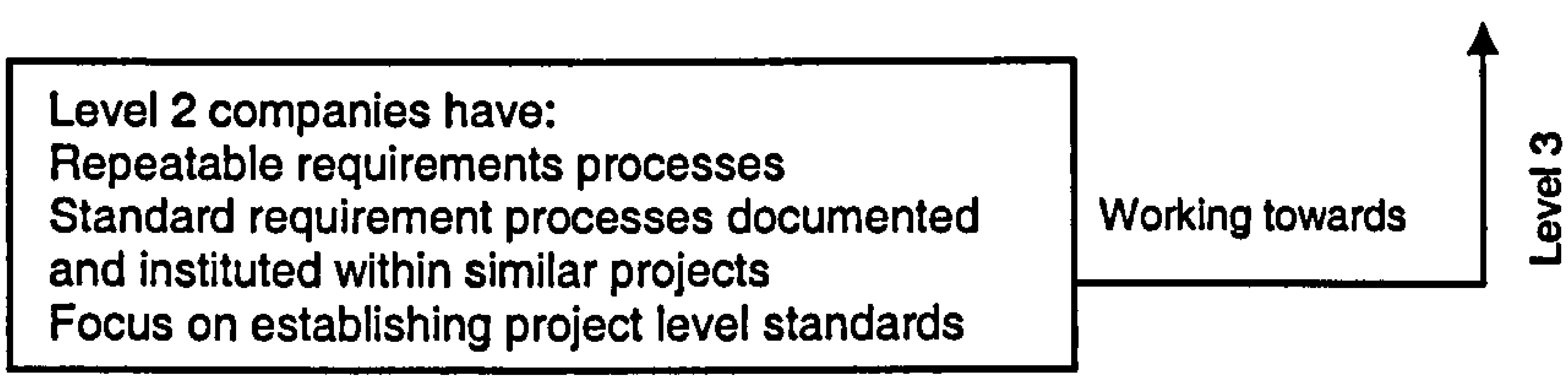
It is not possible to define individual 'process' goals for level 1 companies as these companies operate in their own unique way and depend on people rather than process. Paulk et al (1995) describe success at this level as depending on 'the competence and heroics of the people in the organisation and cannot be repeated unless the same individuals are assigned to the next project". However, a general 'improvement' goal for a company with ad hoc processes is to mature to level 2 where their processes become repeatable.

Level 1 organisations need to work towards developing a disciplined process and need to raise their awareness of their RE process problems. Examining the R-CMM will help managers gain an insight into their RE process and encourage them to buy into the idea of software process improvement. It is likely that managers at this level of maturity will need to prioritise their RE process problems. By definition, this 'ad-hoc' level has no associated 'best practices'. It is at level 2 that the R-CMM addresses the needs of the level 1 companies. To progress to level 2 requires that organisations examine their requirement processes in detail.

According to my empirical study in chapter five, the main requirements related problems level 1 companies are experiencing relate to vague requirements, traceability, defining a RE process, resources, training and skills. As process assessment starts at level 2, however, companies are guided towards examining their current processes prior to implementing new practices.

6.6.2 R-CMM Level 2

Level 2 Goal: To implement a repeatable RE process



(Figure 12 Segment)

Companies at this 'repeatable' Level 2 maturity have established basic project management processes to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications (Paulk *et al.* 1995).

The R-CMM at level 2 maturity can help managers to identify and document their individual RE processes by learning from previous project successes and failures. It introduces controls over processes that may not have been identified as necessary. Managers begin to gain a general overview and can address RE issues associated with individual projects.

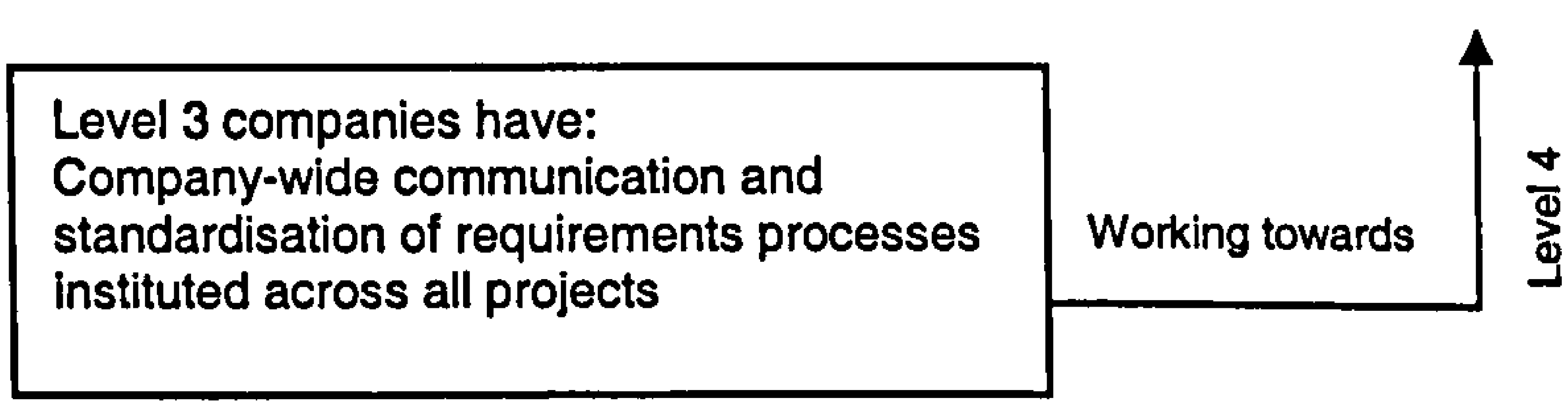
Requirements management is a level 2 key process area (KPA) in the SW-CMM. The R-CMM reflects this by creating a baseline model of RE processes that is built on as a company matures. In the spirit of continuous improvement, a level 2 compliant organisation should be working towards creating a standard and consistent organisation-wide RE process.

My empirical research showed that organisations with a Level 2 capability experience fewer technical problems with their RE process than their Level 1 counterparts. While this suggests that the SW-CMM strategies are working to an extent, my analysis revealed that their organisational problems did not ease off. Problems with communication remain a major problem along with staff retention. For example, my study in chapter 5 showed that technical difficulty for Level 2 companies centred on complex requirements, requirements growth and undefined processes. The R-CMM supports these concerns at this level of maturity.

I present a detailed example of the Level 2 R-CMM in the next chapter.

6.6.3 R-CMM Level 3

Level 3 Goal: To implement a defined RE process



(Figure 15 Segment)

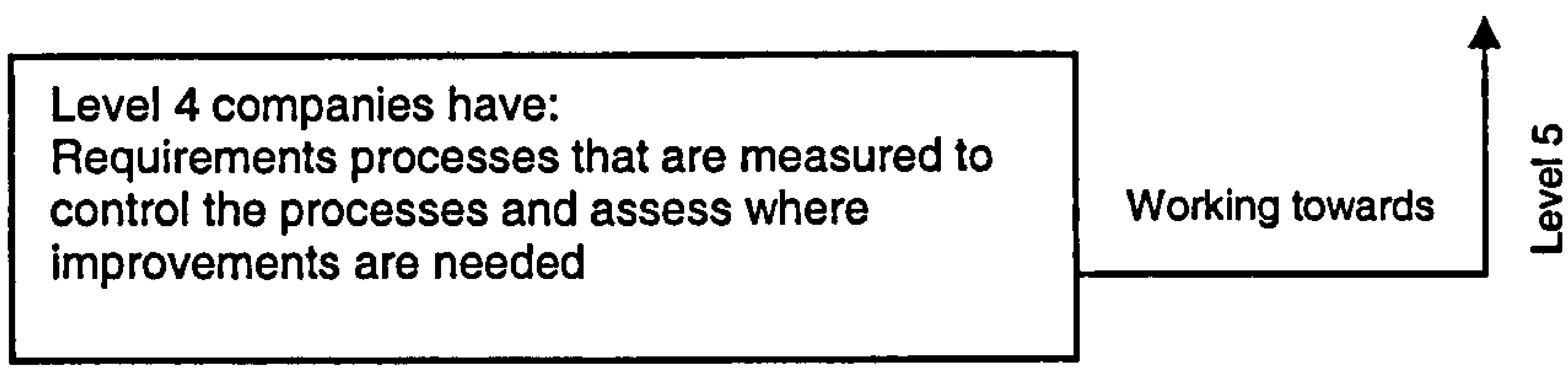
Level 3 R-CMM co-ordinates the standard requirement processes that were established at level 2. The focus shifts from project based processes towards creating company-wide, organisational standards and visibility. All projects now use a documented and approved version of the organisation's process for developing and maintaining software (Paulk et al. 1995) p.193). With these processes in place, management has an increased ability to see and control RE activities.

My empirical research showed that level 3 companies had increased control over their technical RE problems, but saw little improvement in managing their organisational processes. Level 3 companies are most concerned with user understanding of requirements, internal communication and external communication.

The issues raised by practitioner groups at this level of maturity are reflected in the level 3 R-CMM.

6.6.4 R-CMM Level 4

Level 4 Goal: To implement a managed RE process



(Figure 15 Segment)

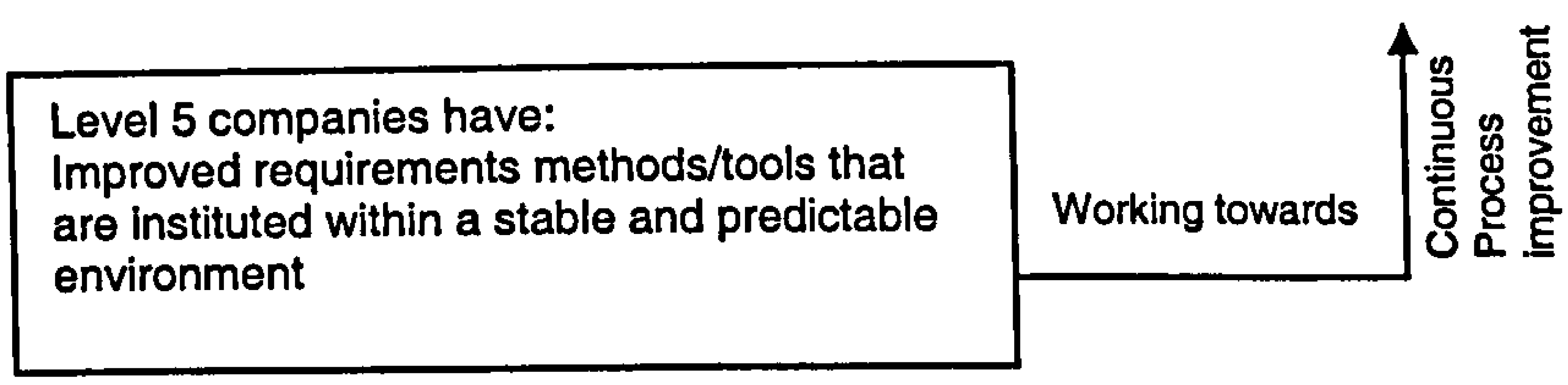
Companies at this 'managed' level 4 maturity are in a position to collect detailed measures of the software process and product quality. Both the software process and products are quantitatively understood and controlled using detailed measurements (Paulk *et al.* 1995).

At this level of maturity, the R-CMM is guided primarily by the SW-CMM. The sample in my empirical study did not produce sufficient data to justify introducing new processes at level 4 maturity. I therefore rely on the SW-CMM and the RE literature to specify activities that focus on the RE process. The R-CMM reflects the SW-CMM focus on measurement at level 4 RE process maturity by introducing quantitative RE quality goals. Examples of measurement data include: effectiveness of RE training; and number and severity of defects found in the software requirements (Paulk *et al.* 1995). The literature shows that both organisational and technical aspects of the RE process can be measured, for example, Nuseibeh and Robertson introduce methods for measuring requirements testability, relevance, completeness, consistency coherence, traceability and satisfaction. "A requirement is regarded as "measurable" if there is an unambiguous way of determining whether a given solution fits that requirement" (Nuseibeh and Robertson 1997). Other examples of improvements gained through measuring requirements and RE processes are noted in (Gresse and Briand 1998; Hammer *et al.* 1997; Lauesen and Vinter 2001; Lavazza and Valetto 2000; Loconsole 2001).

My empirical analysis of level 4 RE process needs are drawn from a small sample of 3 focus groups within one company. I therefore use the results to suggest RE process areas that may need support. For example, the trend to manage technical requirement problems with increased maturity continues as no technical RE problems were reported at this level. However, organisational problems remain a problem, despite the general increase in process capability.

6.6.5 R-CMM Level 5

Level 5 Goal: To implement an optimising RE process



(Figure 15 Segment)

Companies at this 'optimizing' level continually improve their processes through quantitative feedback from the process and from testing innovative ideas and technologies (Paulk *et al.* 1995). Companies moving up from Level 4 to Level 5 should have a wealth of metric data to manage the course of a process (Christie 1999). This creates an environment where elements of processes can be confidently modified. New methods to improve the RE process can be continually tried in a controlled manner.

My empirical study did not include an organisation with a Level 5 software process capability, which is not surprising as there are only a few companies in the world that have reached this level of maturity (SEI 2002). I therefore look to the SW-CMM for maturity characteristics and refer to the RE literature for complimentary best practices. The R-CMM at this high level maturity is therefore a distillation of RE features from the SW-CMM and the literature.

In a fully mature development organisation, the causes of escaped defects are used to improve not only the software requirements specification, but also the review and inspection processes (Smith 1998).

6.7 Conclusion

In this chapter I have shown the initial stages involved in developing a RE process improvement model. The R-CMM breaks away from a linear view of the requirements lifecycle to create a model that is goal and problem driven. The SW-CMM framework is transformed into a simplified model that relates goals and problems to individual RE practices. The result is a specialised, cohesive and comprehensive model that reflects RE key processes at incremental levels of capability.

This chapter also presented generic rules that underpin the model building process. The criteria outlined drive the rest of model development and create a basis for evaluating how well the model meets my objectives. The transparency into the model building process provides a foundation for the next stage of development where individual RE processes and assessment techniques are defined.

Chapter Seven: Defining processes in the R-CMM – a bottom-up approach

7.1 Introduction

This chapter progresses the theme of model development introduced in the preceding chapter. This chapter presents a finer-grained view of the R-CMM where processes are defined and sourced. A goal question metric (GQM) approach is adapted to act as a link between the generic 5-level framework introduced in chapter six, and the measurement of individual processes presented at the conclusion of this chapter. The main purpose of this chapter is to explain the rationale for the selection of specific processes that populate the R-CMM. Processes are included on the basis that practitioners would benefit from monitoring and measuring these processes as a first step to RE process improvement. This bottom-up view of development gives a further perspective on how to create a stable RE process. Research indicates that in order to progress to higher levels of process capability it is essential that these base-line activities are considered.

This chapter is organised as follows: The aim of the study is given in section 7.2, where model development continues to comply with the criteria set in the previous chapter. This chapter aims to present the R-CMM in enough detail to allow for an evaluation of how well the model meets the criteria. Processes become the focus of the model as derived from my empirical study and the literature. Section 7.3 presents the level 2 RE model component where processes are presented within a goal question metric paradigm. Twenty processes are incorporated into the model to form a base-line for RE process capability. It is in this section that goals, questions, processes and metrics are defined. In section 7.4 the importance of the assessors and participants is explained. Processes are extended into detailed guidelines in section 7.5. A detailed model is presented that is also based on a goal question process metric approach. I conclude this chapter in 7.6 with a summary of the study.

7.2 Aim of this study

The aim of this study is to develop one level of the R-CMM in enough detail to allow for an extensive evaluation. Rather than develop all 5 maturity levels of the R-CMM simultaneously, I seek guidance on how well one maturity level of the R-CMM meets my criteria. I focus on the level 2 process maturity as it is this level on which the higher levels of maturity depend. The level 2 model component should guide practitioners to:

a) identify RE processes
b) define RE processes
c) recognise RE process problems
d) assess and agree requirement improvement priorities
e) relate RE process problems to requirement improvement goals
f) relate requirement improvement goals to the software process as modelled in the SW CMM guidelines and activities

Satisfying these aims will ensure that RE processes are identified and included in company goals within a SW-CMM framework.

7.2.1 Defining processes in the R-CMM

Defining processes is recognized as a critical element in software process improvement, yet to be useful a model must be clear and a simplification of the complex world it is modelling. To keep the presentation clear and useable, the R-CMM links processes and maturity levels, but is not an exhaustive representation of the RE process. Processes are included that are considered key to a successful requirement process as based on the research.

Processes are given a structure by coupling them to the SW-CMM at incremental levels of process maturity. Disciplined and structured processes start at level 2. Therefore, levels 2, 3, 4 and 5 in the R-CMM contain processes that create a pathway to high maturity. The processes that populate the R-CMM are one or more of the following:

- a solution to recurring RE process problems raised by practitioners in my empirical study
- a RE-related best practice in the SW CMM
- a recurring theme in the RE literature

Having created a framework, the specialist R-CMM is built up through abstracting data from my empirical study and literature review.

7.2.2 RE process issues raised in the empirical study

Problems raised in my empirical research are viewed in two categories: organisational RE problems and technical RE problems (see Appendix J for a breakdown). Processes included in the R-CMM directly address the problems raised in both these categories.

The specific contents of the R-CMM were driven by RE process data collected in the empirical studies in chapters four and five as published in (Beecham et al. 2003e; Hall et al. 2002a; Hall et al. 2002b). My findings suggest that while there is a significant association between SW-CMM maturity and diminishing technical problems, organisational RE process problems appear untouched by the improvement program. This finding leads me to account for the 'organisational' RE process problems separately to the 'technical' RE process problems. This will ensure that the more difficult 'organisational' processes are not overlooked in the R-CMM.

7.2.3 RE best practices suggested by the literature

By harnessing solutions in the RE literature and relating them to the SW-CMM framework the R-CMM builds on proven work of experts in the field of software engineering. The work of RE experts is used to define technical RE processes. While studies on qualitative aspects of software improvement are also included to add an organisational perspective to the R- CMM, e.g. (Hofmann and Lehner 2001; Perry et al. 1994; Smith 1998).

The literature is rich in suggestions for RE process improvement. However, these recommendations can conflict with each other despite being founded on empirical studies. Bach (1999) advises not to rely on the literature alone to guide practitioners and asks that any advice be 'opened up' to include empirical backing. I have done so by using the literature in conjunction with my own findings to support the best practices within the SW-CMM.

7.3 The R-CMM at level 2 process maturity

The following section involves an analysis of level 2 RE process capability introduced in the previous chapter.

The SW-CMM characteristics, my empirical work and the literature combine to define a level 2 RE process model (see Appendix J for process definitions). Figure 17 is a detailed representation of the level 2 R-CMM where processes are introduced. Figure 17 shows that the capability of Level 2 processes are defined through goals and questions. The rationale for this approach is given in the next section.

One of the objectives of this study is to produce a model of RE processes that is easy to use. I therefore aim to keep the number of processes represented to a minimum. However, some of the advantages of having a concise model are lost if the processes are ambiguous. For example the compound process 'P11', as given in Figure 17, introduces ambiguity into the model. A clearer presentation therefore might be to list the two activities identified in process P11 separately.

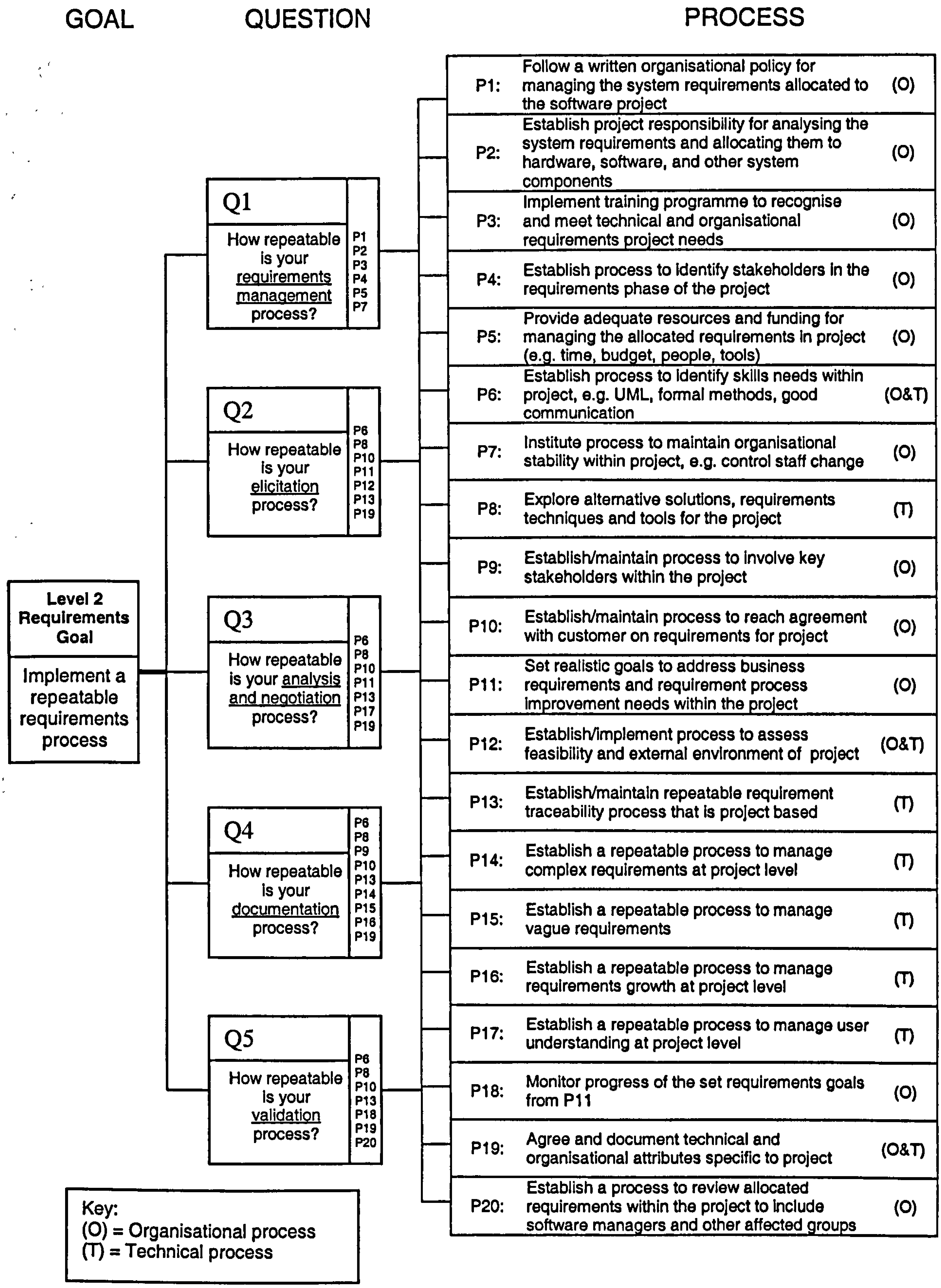


Figure 17: Level 2 R-CMM

7.3.1 The R-CMM Goal Question Process Metric focus

I adapt the Goal/Question/Metric (GQM) paradigm (Basili and Romach, 1988) to be used in the R-CMM. Figure 17 shows how the level 2 goal provides a focus for the model. Processes are modelled separately as identifying and defining processes within a maturity framework is the essence of the R-CMM. This approach is also an ideal way to bring together both technical and organisational needs of a company. The process element directly addresses the needs of a business. For example, the process focus gives a visibility into what is required to improve RE quality (e.g. improved traceability) and can combine these technical processes with those that relate to organisational processes such as resourcing, time-scales and cost that are equally important to business (Solingen and Berghout, 1999).

Figure 17 gives an example of how the organisation sets an improvement goal and how this goal is decomposed through a series of questions that relate to given processes. Figure 18 also shows how the R-CMM supports continuous improvement as advocated by Deming (1982) and Humphrey (1989).

All 5 levels of process capability will follow the improvement cycle presented in Figure 18, where the SW-CMM maturity characteristic (or goal) noted in Figure 17 is decomposed to relate to five requirements phases. Figure 18 shows how a 'process' element has been added to ensure that the required focus is given each of the activities listed.

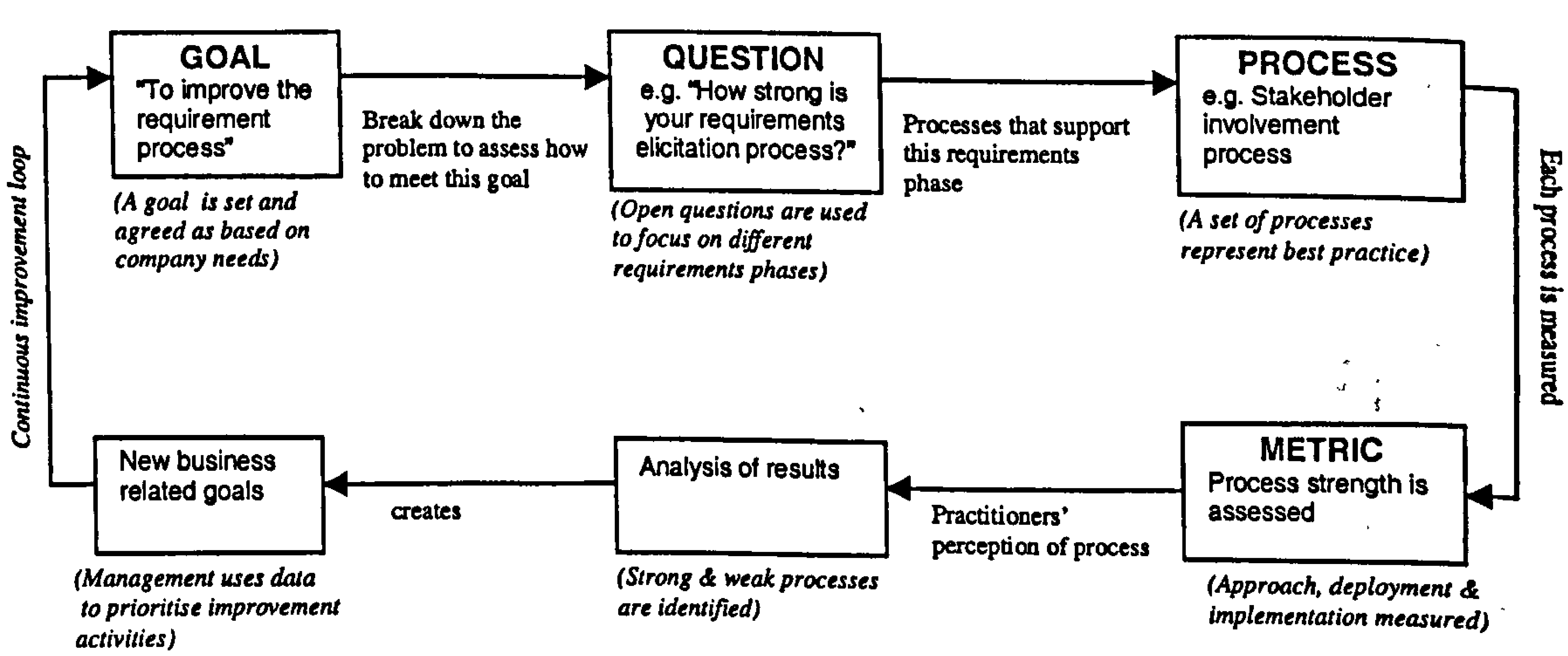


Figure 18: Example of continuous process improvement

This goal, question, process, metric cycle is defined as follows:

- R-CMM Goal

High-level goals are given for each maturity level to work towards. The SW CMM sets the maturity goals for each level of RE process maturity. Figure 17 shows the level 2 goal is to implement a "repeatable" RE process. This level 2 goal is taken directly from the SW-CMM where repeatable software processes are established. "Implement a repeatable RE process" is a refinement of this general goal. Companies who have few controls over their RE processes need to work towards instituting the baseline activities introduced at this level.

The R-CMM differentiates between (a) the high level (top down) goals as given in Figure 15 that provide a focus for each maturity level and (b) the individual business related goals featured in Figure 18. This distinction is made to ensure that the improvement effort is driven primarily by *business* goals and not maturity level goals (Wiegers, 1998a). To meet these aims, the R-CMM guides practitioners towards identifying processes that enable them to achieve their specific requirement improvement goals as implied in Figure 18.

- R-CMM Questions

Questions are used to interrogate whether processes are in place to comply with maturity level characteristics. For example, Figure 17 shows that assessing whether a level 2 RE process goal has been achieved requires addressing five questions. For completeness and ease of use, the five questions relate directly to recognised requirements phases: management; elicitation; analysis and negotiation; documentation; and validation (Dorfman and Thayer 1997; Pressman 2001). Each maturity stage will view the RE process in these phases for consistency and ease of implementation. The purpose of this phased view is to help practitioners relate individual RE processes to practices. Even if companies do not view their RE process in these defined phases (as indicated by the literature), it is likely that the processes within these phases of development are understood. However the questions do impose a contrived order on the data, where processes are grouped together to help identify where general strengths and weaknesses are in the RE process. Without this phased view of RE, it might be difficult for practitioners to know where to start.

Questions are purposefully designed to be 'open' to guide users to investigate whether the goal is being met. Providing answers to these set questions will help managers gauge whether progress is being made towards meeting the goal (Pfleeger, 1995). Questions are quantifiable and have three sub-goals: they are defined; the quality perspective of interest is defined; and feedback from using this process relates to the quality perspective of interest (Basili and Romach, 1988). To meet these sub-goals the R-CMM questions are clearly defined as in (Beecham *et al.*, 2003b); each question relates directly to the goal; and each question leads into the solutions provided by the processes. Output from answering questions will be assessed and fed back into the model to define new goals. For example, "How repeatable is your elicitation process?" contributes directly to the level 2 goal of creating a repeatable RE process and relates to quantifiable processes.

A RE process can apply to more than one requirements phase, and this relationship is modelled in the R-CMM. For example the process P13 "Establish/maintain a repeatable traceability process that is project based" relates to the elicitation, analysis, documentation and validation phase of producing requirements. Whereas the process P4 "Establish process to identify stakeholders in the requirements phase of the project" is set up in the requirements management phase and therefore relates to this initial stage only. By introducing processes in this way the R-CMM bridges the gap between a traditional, structured 'lifecycle' view and the more fluid process view of RE as "requirements are developed iteratively based on feedback about baseline documents and evaluation of system prototypes"(Peters and Pedrycz, 2000).

- Processes: the substance of the R-CMM

The R-CMM 'process' dimension places RE processes in context with goals and questions at different levels of process maturity. Each process represents best practice and addresses problems highlighted in my empirical research. How capable a company is in implementing the process relates to a level of requirements process maturity. However, if a company decides not to include some of the recommended processes they should first run an objective assessment to determine its importance. This is because viewing a process in isolation can uncover hidden weaknesses.

Modelling processes separately therefore allows companies to examine and prioritise their RE process improvement activities. Also, viewing processes independently eases the transition from a descriptive process (that addresses 'what' should be done to improve processes) to a more applicable prescriptive process (that shows 'how' the process can be implemented). Definitions of all the processes presented in the level 2 model are given in Appendix L.

Processes reflecting maturity level characteristics are identified for each question. For example, Figure 17 shows that, overall, I have identified twenty RE processes that my study suggests are key to establishing a Level 2 capability. Organisational and technical processes are separated to ensure that organisational processes are given a similar focus to the technical processes. For example "identify stakeholders", "involve stakeholders", "identify skills needs" represent organisational processes while "establish and maintain a RE traceability process", and "implement a process to address complex requirements" represent technical processes. Each level of maturity has a unique set of recommended processes.

- Motivation for including processes in the R-CMM

Twenty processes were selected as key to baseline RE needs at a project level. The primary motivation source for selecting each process is shown in Table 25.

The SW CMM motivated category (category 1 in Table 25) covers many of the 'organisational' activities within the RE process. From my empirical study I conclude that organisational issues are causing practitioners more problems than technical issues. The emphasis the SW-CMM places on 'managing' the RE process is therefore warranted. However, the processes included in categories 2 and 3 indicate that the SW-CMM requires enhancement.

Table 25: Motivation for including RE processes in the R-CMM at level 2 maturity

Source	Level 2 RE Processes
1.SW CMM	P1: Follow a written organisational policy for managing the system requirements allocated to the software project P2: Establish project responsibility for analysing the system requirements and allocating them to hardware, software, and other system components P5: Provide adequate resources and funding for managing the allocated requirements in the project P20: Establish a process to review allocated requirements within the project to include software managers and other affected groups
2. Empirical Research	P3: Implement training programme to recognise and meet technical and organisational RE needs within the project P4: Establish process to identify stakeholders within the project P6: Establish process to identify skills needs within project, e.g. UML, Formal methods P7: Institute process to maintain organisational stability within project, e.g. control staff change P10: Establish/maintain process to involve key stakeholders in requirements phase of project P13: Establish/maintain repeatable requirement traceability process that is project-based P14: Establish a repeatable process to manage complex requirements at project level P15: Establish a repeatable process to manage vague requirements at project level P16: Establish a repeatable process to manage requirements growth at project level P17: Establish a repeatable process to manage user understanding at project level P19: Agree and document technical and organisational attributes specific to project
3.Literature	P8: Explore alternative solutions, RE techniques and tools for the project P9: Establish / maintain process to reach agreement with customer on requirements for project P11: Set realistic improvement goals to address problems in the RE process P12: Establish/implement process to assess feasibility & external environment of project P18: Monitor progress of the set requirements goals

Although *all* SW CMM Key Process Areas (KPAs) start with 'goals' there is nothing specific about how companies should identify their own goals based on their own personal RE process weaknesses. Therefore I look to the literature for guidance, e.g. (Davis, 1988; Sawyer *et al.*, 1997; IEEE, 1998) who suggest companies set realistic improvement goals when planning for RE process improvement. Also, all KPAs include the need to assign responsibilities and resources to each activity. We have adapted these practices to relate specifically to the RE process rather than general software development.

- R-CMM metric focus

It is through analysing processes and assessing their strength that a company can determine how well their goals have been met. Measuring the strength of a process will also lead to a better understanding of current practices that in turn will help companies to set realistic project goals (Basili, 1995) and (Madhavji, 1991). The R-

CMM guides companies to measure individual process strengths, as shown in the 'Metric' dimension in Figure 18, whilst retaining a goal focus.

Setting realistic goals means recognising and prioritising which processes need strengthening. The R-CMM employs a tried and tested assessment technique as used by Motorola (Daskalantonakis 1994) to track progress in achieving a high SW-CMM level. Motorola developed the process evaluation as given in Table 26 to allow them to perform internal, incremental assessments as they felt that waiting two years between SEI formal assessments was too long. The generic evaluation questionnaire analyses the company's approach, deployment and application of the process.

Table 26: Process Capability Scoring Matrix

Score	Key Activity evaluation dimensions		
	Approach	Deployment	Results
Poor (0)	• No management recognition of need • No organisational ability • No organisational commitment • Practice not evident	• No part of the organisation uses the practice • No part of the organisation shows interest	• Ineffective
Weak (2)	• Management begins to recognise need • Support items for the practice start to be created • A few parts of organisation are able to implement the practice	• Fragmented use • Inconsistent use • Deployed in some parts of the organisation • Limited to monitoring/verification of use	• Spotty results • Inconsistent results • Some evidence of effectiveness for some parts of the organisation
Fair (4)	• Wide but not complete commitment by management • Road map for practice implementation defined • Several supporting items for the practice in place	• Less fragmented use • Some consistency in use • Deployed in some major parts of the organisation • Monitoring/verification of use for several parts of the organisation	• Consistent and positive results for several parts of the organisation • Inconsistent results for other parts of the organisation
Marginally qualified (6)	• Some management commitment; some management becomes proactive • Practice implementation well under way across parts of the organisation • Supporting items in place	• Deployed in some parts of the organisation • Mostly consistent use across many parts of the organisation • Monitoring/verification of use for many parts of the organisation	• Positive measurable results in most parts of the organisation • Consistently positive results over time across many parts of the organisation
Qualified (8)	• Total management commitment • Majority of management is proactive • Practice established as an integral part of the process • Supporting items encourage and facilitate the use of the practice	• Deployed in almost all parts of the organisation • Consistent use across almost all parts of the organisation • Monitoring/verification of use for almost all parts of the organisation	• Positive measurable results in almost all parts of the organisation • Consistently positive results over time across almost all parts of the organisation
Out-standing (10)	• Management provides zealous leadership and commitment • Organisational excellence in the practice recognised even outside the company	• Pervasive and consistent deployment across all parts of the organisation • Consistent use over time across all parts of the organisation • Monitoring/verification for all parts of the organisation	• Requirements exceeded • Consistently world-class results • Counsel sought by others

(Daskalantonakis 1994)

This scheme reflects the SW-CMM focus on

- evaluating a company's commitment towards the practice
- assessing typical activities expected of the practice
- checking that metrics are taken of the process
- checking that metrics are evaluated

(Love and Siddiqi 2000)

The structure of the R-CMM allows for each of the 'phases' to be assessed as shown in Figure 19.

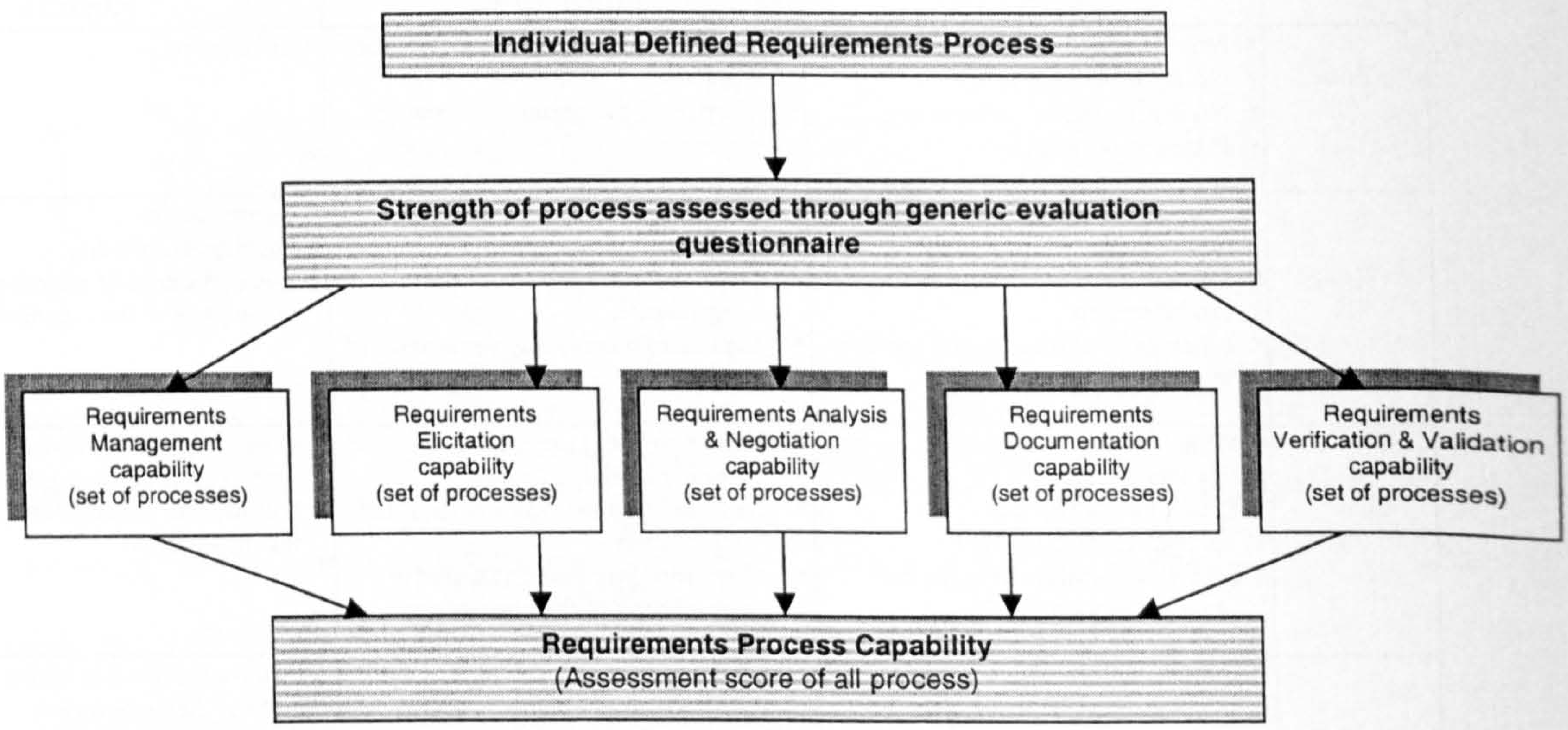


Figure 19: Stages in RE process assessment

The process evaluation form and measurement criteria are given in Appendix M.

7.4 Assessors and participants

Assessment results are often dependent on the subjective interpretation of assessors and are not, therefore, reliable for long-term benchmarking and monitoring (Kauppinen et al. 2002). I take lessons learnt from this assessment study and create a systematic scheme that is an internal assessment based on the sound judgement of those who are using the process. The R-CMM assessment questionnaire is detailed enough to include all essential areas and the results of the assessment should give a realistic account of the current state of the RE process. In line with SEI advice, the R-CMM advocates that when determining who should fill out the questionnaire, individuals are chosen who will provide answers that represent the entire project/organisation/requirements phase (SEI 1996). Choosing a representative cross section of RE stakeholders should also ensure consistent results over time.

The assessment forms the final part of the Goal/Question/Process/Metric paradigm. Metrics are used to quantify how well a process has been approached, deployed and what the results of implementing the process yields. Assessment is viewed as an essential element of process improvement as any process improvement effort should begin with some kind of assessment, to establish a baseline understanding of current practices and problem areas (Wiegers, 1998a). It is only through an assessment that companies can gain a balanced picture of where their current practices need improving. Further, it is only with this knowledge that companies can set realistic RE process improvement goals (Davis, 1988; Sawyer *et al.*, 1997; IEEE, 1998). Practitioners need to identify their own specific reasons for wanting to improve their performance and the assessment will lead companies to look at their current practices and set realistic goals when planning for further RE process improvements.

An example of a how RE processes are assessed in the R-CMM is given in Appendix M. A study of process assessment is given in (Beecham *et al.*, 2003a).

7.5 The R-CMM guideline

A criticism of the SW CMM is that it is too descriptive and does not provide sufficient examples and specific guidelines to help companies with their process improvement activities, e.g. (Lauesen and Vinter, 2001; Potter and Sakry, 2001). By taking key RE processes and extending them into detailed guidelines the R-CMM features specific RE processes that in turn can be measured. Every process is defined in detail through references to prescriptive solutions in the literature. This takes on the modelling heuristics of breaking down a complex high-level problem (i.e. the requirement process) into sub-processes and then building on these sub-processes in an iterative fashion (Srinivasan and Te'eni 1995).

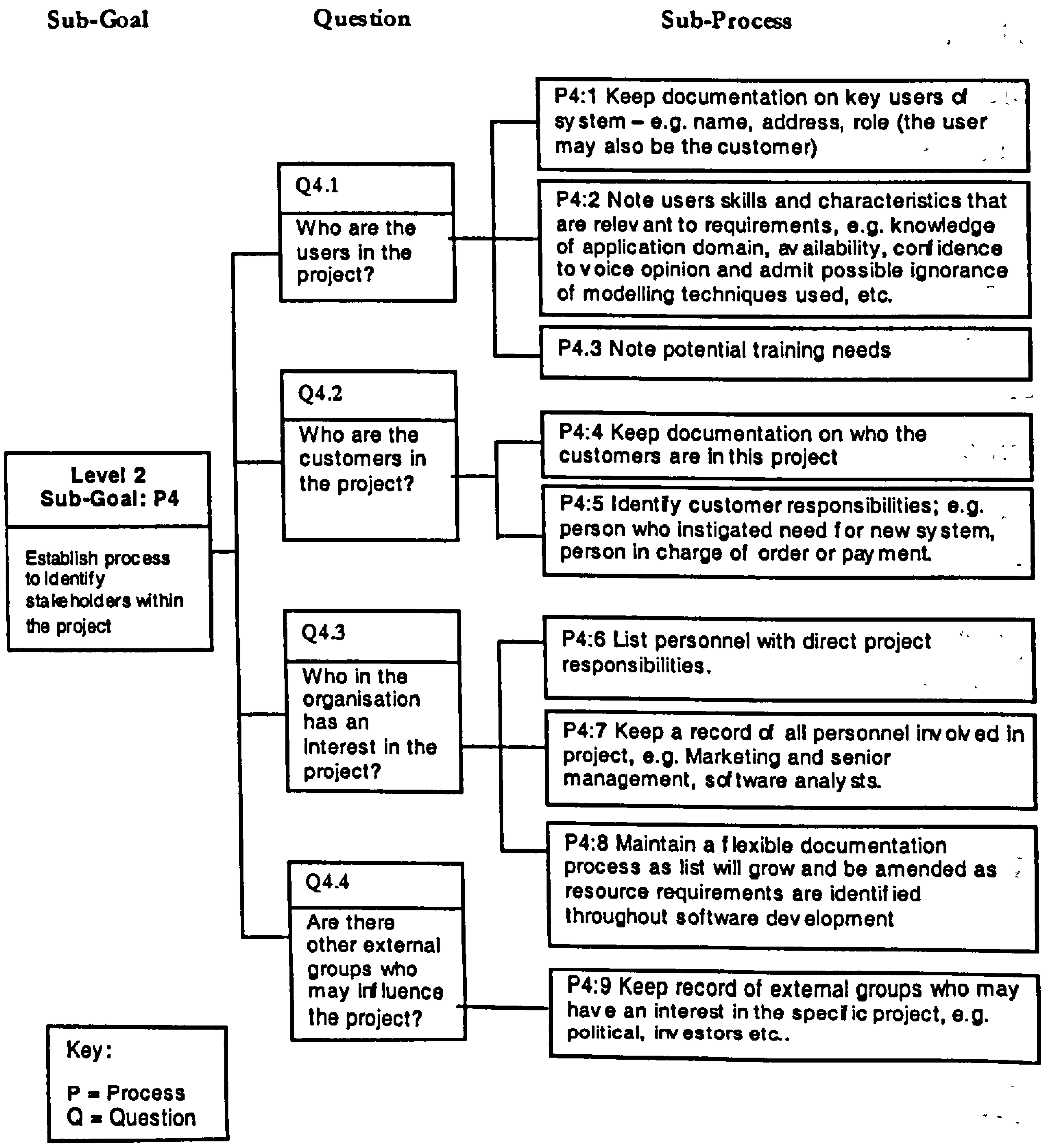


Figure 20: Guideline example of a Level 2 RE Process

In this process refinement, goals become more defined. Looking at Figures 17 and 20, it is possible to see how the goal to "establish a process to identify stakeholders within the project" is derived from the higher level model. The guideline model in Figure 20 also retains the GQM approach to improvement. Guidelines move away from the descriptive model design as they prescribe lower level practices that are needed to achieve improvement goals. They are based on an analysis of the RE literature. The example given in Figure 20 is based on the work of Sommerville and Sawyer (1997) and is supported by findings in (Hall et al., 2002b). The work of (Boehm 2001; Standish Group 1995; Thayer and Dorfman 1990; El Emam et al. 1996 and Hofmann and Lehner, 2001) also highlight the importance of this process. The guideline gives examples of how processes might be implemented but retains a descriptive rather than prescriptive focus. There is a balance between producing a model that is too prescriptive (that will be very helpful to a few companies) and producing a generic 'descriptive' model that has a more universal application.

The technique for measuring processes can be extended to assess the strength of sub-processes as given in the Figure 20 example, should a finer grained analysis be required.

7.6 Conclusion

This chapter describes how I adapt the SW-CMM to focus on RE processes. I demonstrate this applied improvement methodology through a series of model components that isolate the requirements phase of software development. The chapter focuses on level 2 R-CMM to show how practitioners are guided towards recognising baseline RE processes.

The processes defined in the Level 2 R-CMM work together to produce a baseline structure for companies to consider within their software development activities. The clear definitions given in the guidelines will help with process implementation. Also, using the GQM approach will guide practitioners towards improving and managing the RE process through recognising the specific needs of the organisation. The R-CMM therefore guides users to create specific goals based on their business needs.

The model directs practitioners to examine their RE process in a systematic and detailed way. The R-CMM includes some SW-CMM best practices together with additional RE processes that are outside the scope of the SW-CMM. The study shows how processes included in the R-CMM that are not explicitly modelled in the SW-CMM are included on the basis of meeting the needs of practitioners in my empirical study, as well as taking best practices from the literature.

This chapter develops the high level view of the RE process given in the previous study into a more useful tool. The aim is for the R-CMM to support both practitioners and researchers in the field of process improvement and RE. This chapter also shows how the R-CMM guides and prompts practitioners through the diverse processes involved in RE process improvement.

It is also intended that detailing the actual processes involved in developing the model will provide a foundation for future development in the area of RE process improvement. The R-CMM as presented in this chapter and the previous chapter is now evaluated against my original success criteria. The detail given that covers general top down development together with detailed processes provides enough information to allow for an in-depth analysis of the model's strengths and weaknesses.

Chapter Eight: Validating the R-CMM

8.1 Introduction

This validation study is the culmination of the work involved in developing a process maturity model that focuses on RE. I now reach the stage in model development where I need independent feedback to validate how well the R-CMM meets my objectives. I perform this validation through involving a group of SPI and RE experts in examining the R-CMM components and completing a detailed questionnaire. A major part of this study is devoted to reporting and analysing the results of the expert panel validation questionnaire. Although this validation study concludes this thesis, it is by no means the completion of model development. Rather, the results from this study will provide the impetus for further model development.

In this study I validate whether the motivation for building the R-CMM is justified and whether the model reflects the needs of the software industry (Burnstein *et al.* 1996). I present my validation methodology and report the experts' responses to a detailed validation questionnaire. The study is exploratory and looks at the strengths and weaknesses of this requirements-based software process maturity model at an early stage of development. The main processes involved in validating the R-CMM are:

1. List the criteria for R-CMM development identified during the initial stages of model development (in chapter six, Table 24);
2. Explore alternative methods for testing how the criteria are reflected in the model;
3. Design a validation instrument to test the success criteria;
4. Apply and implement the validation instrument;
5. Present results to the validation instrument;
6. Discuss how the results relate to my success criteria.

Results from this validation phase will impact the continuing development of the R-CMM and constitute the main driver for future work.

Objectives are set at the start of model development to clarify the purpose and rationale for creating the model. Having a clear set of objectives helps to steer model development and creates the criteria against which I now test the model for correctness and completeness (Madhavji 1991). I adapted practices from model validation guidelines to my specific purposes.

This chapter is organised as follows. The choice of survey instrument is discussed in section 8.2. The purpose and results of a pilot study is given in section 8.3. In section 8.4 the expert panel demographic details are given along with the response rate. The responses of the survey instrument are placed in context in section 8.5 where the experts' perception of the SW-CMM and the state of RE processes in practice is explored. Section 8.6 presents the results of the questionnaire that relate to the seven success criteria and includes the experts' overall impression of the R-CMM. Section 8.7 gives a summary of the results and highlights perceived R-CMM strengths and weaknesses. In section 8.8 the findings of the study are discussed in relation to the needs of the software industry. I conclude this chapter in section 8.9 with a summary and directions for further work.

8.2 Validation instrument

In order to validate the model I need to replicate questions directly associated with my model criteria. I therefore choose the questionnaire as my primary data collection method as it is best suited to the nature and type of data that I need to analyse (Rodeghier 1996). Results from this questionnaire are used, with caution, to indicate possible strengths and weaknesses within the model. Results are also used to consider the wider implications of the experts' attitude to the SW-CMM and RE process which are likely to be of interest to research and development. A copy of the expert panel validation questionnaire is given in Appendix E.

8.2.1 Questionnaire design

The primary purpose of this validation questionnaire is to establish how well the R-CMM meets the success criteria outlined at the start of development. Questions are grouped together to explore strengths and weaknesses of individual model components. Questions are designed to explore several model components which are grouped together to satisfy each success criteria as shown in the entity relationship diagram in Figure 21.

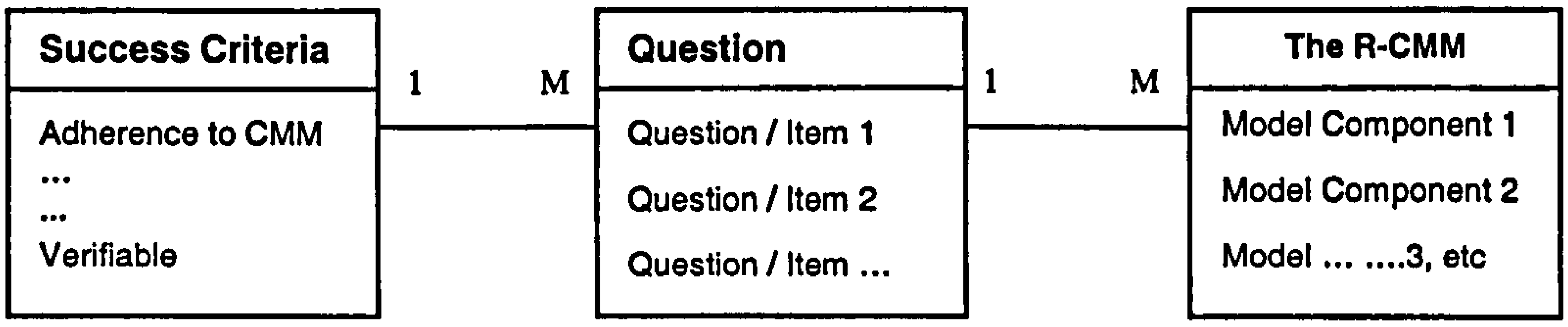


Figure 21: Relationship between Success Criteria, Questionnaire Design and R-CMM

The entity relationship model in Figure 21 shows that multiple items are combined in the questionnaire to test each success criteria and help average out possible errors in single item responses (Dybå 2000).

To test whether the items do indeed relate to the success criteria two separate researchers categorized the items according to my success criteria. I test the reliability of these categorizations through the Cohen's kappa (κ) statistic where agreement between the evaluations of two raters (rating the same object) is measured (SPSS 1999). The extent of this agreement is illustrated in Table 27, the numbers in bold show where the two researchers agree.

Table 27: Two researchers' classification of 32 items

		Researcher 1							
	Criterion	A	B	C	D	E	F	G	*Total*
Researcher 2	A	**4**	1	0	0	0	0	0	*5*
	B	0	**4**	3	0	0	0	0	*7*
	C	0	0	**3**	0	0	0	0	*3*
	D	0	0	0	**4**	0	0	0	*4*
	E	0	0	0	0	**5**	0	0	*5*
	F	0	0	0	0	0	**7**	0	*7*
	G	0	0	0	0	0	0	**1**	*1*
	Total	*4*	*5*	*6*	*4*	*5*	*7*	*1*	*32*

Criteria Key:
A = Adherence to CMM
B = Scope
C = Consistency
D = Understandable
E = Ease of Use
F = Tailorable
G = Verifiable

The kappa statistic for this inter-rater reliability test is .85 which indicates an almost perfect agreement (Landis and Koch 1977), (Cramer 1997). I am therefore confident that the groups of multiple items do indeed relate to the associated criteria. However, analysis of the four disagreements resulted in two items being moved to a more suitable criterion. The test was therefore of use in confirming a high level of agreement as well as highlighting individual inconsistencies in item classification.

8.3 Pilot study

In order to uncover potential problems in the design and application of the questionnaire I ran a pilot study involving five researchers in the fields of RE and SPI. The test involved an assessment of the respondents' level of understanding, level of knowledge, level of difficulty in responding and level of relevance to subject area (Berry and Jeffery 2000). The feedback from this pilot study led to the following changes:

- Level of understanding: I created a web-page to include further definitions and background information relating to the model.
- Level of knowledge: Experts were specifically chosen for their knowledge of either the RE processes and/or the SW-CMM. A covering letter explained that they were not expected to have knowledge in both areas. They could give 'no opinion'/'don't know' response to any model related question.
- Level of difficulty. The pilot study highlighted areas that were difficult to answer as in some cases the participants did not have the required expertise. The problems arose more through this lack of knowledge than questions being ambiguous. I included a page at the end of the questionnaire for participants to note their queries.
- Level of relevance. None of the pilot study participants questioned the relevance of the questions. I had previously removed unnecessary questions relating to company demographics.

As I piloted the whole questionnaire, I was able to assess the level of time commitment required to complete the questionnaire which was approximately one hour.

8.4 Expert panel response rate and demographics

Experts were targeted in academia for having published work on RE and/or SPI, whereas industrial experts were selected for their experience in the field of RE and/or implementing improvement programs. I invited twenty-seven experts to participate in validating the R-CMM and twenty-three accepted (representing a take-up rate of 85%). However, twenty completed questionnaires were received representing a final response rate of 87% of experts who accepted my invitation to participate. As we are unable to confirm the reason for the non-participation of the 13% of experts who agreed to participate and did not return their questionnaires, some likelihood of bias is present. As the sample is not random I cannot claim that a response rate of over 80% is unlikely to bias survey results (SPSS 1996). However, the twenty experts who participated in the validation represent a good spread of knowledge as shown in Table 28:

Table 28: Distribution of expertise in R-CMM validation panel

	Role			
Field of expertise	Practitioner	Academic	Practitioner & academic	Total
SW-CMM only	1	0	0	1
RE only	6	4	1	11
SW-CMM and RE	3	3	2	8
Total	10	7	3	20

Table 28 shows that most experts categorised themselves as having a good or expert knowledge of RE (95%), whereas only 45% have a good or expert knowledge of the SW-CMM. Appendix D gives a breakdown of the R-CMM validation panel who agreed to be named.

8.5 Response categories

I sent each member of the expert panel a questionnaire and an accompanying R-CMM documentation booklet (see Appendices E and F). The accompanying documentation comprises a set of model components that guide the expert from a high level framework view through to a detailed guideline view of the R-CMM. It also gives an overview of the purpose of the R-CMM and what it is endeavouring to represent. For more details see (Beecham *et al.* 2003a; Beecham *et al.* 2003b; Beecham *et al.* 2003c; Beecham *et al.* 2003e).

The analysis of questionnaire responses falls in three categories:

1) establish experts' view on the established SW-CMM as a process improvement model;
2) note how strongly the expert feels about the need for RE process support; and
3) measure the experts' perception of how well the model complied with the success criteria.

The first two points address broad issues associated with evaluation and validation, whereas the third point specifically relates to the validation of the R-CMM. Prior to analyzing responses that relate to my success criteria we need to establish how experts view the SW-CMM and the current state of the RE process as this may influence how they view R-CMM.

8.6 Questionnaire results

This section presents expert responses to key questions in the questionnaire. The numbering of the questions in the tables reflects the order in which they appear in the questionnaire. The term framework and model are used interchangeably.

8.6.1 Attitude to the SW-CMM

The experts' view of the SW-CMM is of interest as there are implications for inherited strengths and weaknesses in the R-CMM. Also the experts' view of the SW-CMM may influence how they respond to similar questions relating to the R-CMM. Table 29 presents expert responses to questions that relate to the SW-CMM.

The confidence limits in Table 29 show that using the SW-CMM is likely to be met with some resistance as support for the SW-CMM is not universal. Also, it is likely that the R-CMM is inheriting some weaknesses as perceived by the experts. Of particular concern is the 43% disagreement that the SW-CMM is a model that reflects current best practices, shown in item 16.

Table 29: Expert attitude to the SW-CMM

Questions	'No opinion' responses	No. of valid responses	Critical responses	Supportive responses: Observed	Supportive responses: %	Confidence interval 95%: LL, UL
13. The SW-CMM clearly defines software process activities	2	18	4	14	Agree = 56 Str agree = 22 *Total= 78*	[55, 91]
14. The SW-CMM's 5 stage framework helps companies to prioritise process implementation	2	18	4	14	Agree = 67 Str agree = 11 *Total = 78*	[55, 91]
15. The SW-CMM framework can be tailored to suit a company's specific needs	6	14	3	11	Agree = 64 Str agree = 14 *Total= 78*	[52, 92]
16. The guidelines in the SW-CMM represent current best practice in software	6	14	6	8	Agree = 57 Str agree = 0 *Total = 57*	[33, 79]

There is general enthusiasm for the SW-CMM as a SPI model with 78% support for most items in this category. However within this group of responses there were some criticisms with 22% or more experts being critical of the SW-CMM. In order to assess whether individual perceptions of the SW-CMM are carried through to the R-CMM validation, I divided the participants into those who were critical of the CMM ($n = 9$) and those who were not ($n = 9$). An exact chi squared test and a Mann-Whitney U test were performed to compare how the two groups responded to seven key R-CMM questions. The chi-squared test results shows no significant association between row and column variables and the Mann-Whitney U test results indicate that, in general, the two sets of responses are drawn from identical populations (n critical = 9, n supportive = 9, $p > 0.05$ in six out of the seven key questions). Therefore, experts who are critical of the SW-CMM and experts who are supportive of the SW-CMM are giving similar responses to R-CMM related questions.

8.6.2 The RE process as a problem

Table 30 shows a consensus amongst the experts that in general the RE process is in need of further improvement as companies continue to experience problems with this software development phase. Only one expert believes that RE do not cause more problems in development than any other software engineering activity.

Table 30: Expert opinion of the RE process

Questions	'No opinion' responses	No. of valid responses	Supportive responses Observed	%	Confidence interval 95% LL, UL
5. There is evidence to suggest that companies have problems with their requirements process	0	20	20	Agree = 20 Str agree = 80 *Total support = 100*	[84, 100]
6. It is likely that the requirements process leads to more problems in development than other software engineering activities	0	20	19	Agree = 45 Str agree = 50 *Total support = 95*	[76, 99]
7. In general, the requirements process is in need of improvement	1	19	19	Agree = 32 Str agree = 68 *Total support = 100*	[83, 100]

8.6.3 R-CMM success criteria

The seven success criteria as detailed in Table 24 (in chapter 6) are tested through responses to multiple items. Results of each success criteria are listed and discussed in this section.

Criteria 1: Adherence to SW-CMM

Questions in Table 31 test how well the R-CMM adheres to the SW-CMM structure and concept.

Table 31: An indication of R-CMM adherence to the SW-CMM

Questions	'No opinion' responses	No. of valid responses	Supportive responses Observed	%	Confidence interval 95% LL, UL
32. How well do the questions (based on 5 requirements phases) relate to the Level 2 (SW-CMM) goal?	1/20	19	17	(3) = 37.5% (4) very = 52% *Total = 89.5%*	[69, 97]
72. The assessment method retains the CMM level concept	6/20	14	14	Agree =86% Str Agree = 14% *Total= 100%*	[78, 100]
76. How well does the new requirements framework retain the SW-CMM concept?	6/20	14	14	(3) = 50% (4) very = 50% *Total= 100%*	[78, 100]

Looking at Table 31, all participants who gave a valid response were in agreement that the framework (Item 76) and the assessment method (Item 72) retain the CMM concept. This model integration should avoid redundant activities that may occur if the two models had separate maturity level concepts. The four participants who rated themselves as having no previous knowledge of the SW-CMM all gave a 'no opinion' response when asked about the framework retaining the SW-CMM concept. This result gives us confidence in how the experts are answering the SW-CMM related questions.

Support is slightly reduced as experts examine how well the five phases of the RE process adhere to SW-CMM maturity goals (Item 32). However, as all items used to test the SW-CMM adherence criteria include 80% support within their confidence limits, this area is not of immediate concern.

Criteria 2: Limited Scope

Questions in Table 32 test the scope of the R-CMM to include inclusion of key processes and level of completeness.

Table 32: Expert opinion of R-CMM scope

Questions	'No opinion responses	No. of valid respons es	Supportive responses		Confidence Interval 95%	
			Observed	%	LL	UL
19. How complete is the requirements CMM framework? (asked at beginning of questionnaire when examining high level model)	6/20	16	8	(3) = 29 Very = 29 Total = 58	[33, 79]	
28. How appropriate is the level of detail in the requirements CMM for an initial guide to the requirements process?	1	19	11	(3) = 47 Very = 11 *Total support = 58*	[36, 77]	
29. How appropriate is it to include organisational processes (e.g. requirements skills audit) and technical processes (e.g. techniques to trace requirements) in one model?	0	20	18	(3) = 25 Very = 65 *Total support = 90*	[70, 97]	
34. How well do questions [the 5 requirements phases] cover all the key activities involved in the requirement stage of software development?	2	18	17	(3) = 61 Very = 33 *Total support = 94*	[74, 99]	
41. Each process relates to requirements engineering activities	0	20	11	Agree = 40 Str agree = 15 *Total = 55*	[34, 74]	
77. How complete is the requirements CMM framework? (asked at end of questionnaire)	5/20	15	10	(3) = 53 Very = 13 Total = 66	[42, 85]	

Table 32a: Expert opinion on the level of information provided

	Too few (1)	Correct Number (2)	(3)	(4)	Too many (5)	No opinion	Total
27. We have divided the requirements phase of software engineering into 20 key processes. Is this a good number or would the model benefit from a more comprehensive list?	0	2	6	10	1	1	20
51. There is sufficient detail in document 3 to guide the user towards recognizing baseline requirements processes	2	4	6	2	1	5	20

Table 32a presents raw scores only because the odd number of response categories cannot be dichotomised.

Testing the scope of the model falls into four categories: process inclusion, number of processes level of detail and model completeness.

- Process inclusion

There is a strong agreement that it is appropriate to include both technical and organisational processes in the model (Item 29), yet there is a critical response to these same processes being categorised as RE activities (Item 41). A McNemar's test (SPSS 1999) shows that there are differences in how individuals respond to the two questions ($X^2 = 5.143$, df = 1, p = 0.016 for two-tailed exact test with N = 20 cases) (Everitt 1992). I am therefore left with a dilemma as to which processes are appropriate to include in the R-CMM.

- Number of processes

"Experience has shown that organisations do their best when they focus on a manageable number of process areas" (Konrad and Shrum 2001). Item 27 indicates that the twenty key baseline processes are slightly too many for this level of abstraction. Alternatively, this response may suggest that the model contains some sub-processes that are not considered key to the RE process as indicated by Item 41. The number of processes included in process improvement models varies. For example, the SW-CMM has only five key process areas at level 2, whereas the Requirements Good Practice Guide (Sommerville and Sawyer 1997) includes 36 guidelines in their more detailed Level 2 process model.

- Process level of detail

The model would possibly be enhanced by giving each process a greater depth of detail as shown through the critical results in Items 28 and 51.

- Model Completeness

I asked how complete the R-CMM high level model is (Item 19) and how complete the model is again at the end of the questionnaire when participants had looked at all the model components (Item 77). Comparing these two items, the slight increase in valid response level and support for the model at the end of the questionnaire suggests that looking at all the model components led to experts gaining a better understanding of the R-CMM. However, it is a difficult question to answer, as the model is not intended to be truly 'complete'; this validation acts as a guide to my further development. However a good level of completeness is confirmed by the five phases covering all activities involved in the requirements phase of development (Item 34). However, this positive response relates to a high and conceptual level of detail.

Criteria 3: Consistency

Questions in Table 33 test whether R-CMM features are consistent.

Table 33: An indication of R-CMM consistency

Questions	'No opinion' responses	No. of valid responses	Supportive responses Observed	Supportive responses %	Confidence Interval 95% LL UL
21. How consistent is the level of detail given within the Requirements CMM	2/20	18	17	(3) = 50 Very = 44 *Total = 94*	[74, 99]
40. All Key processes are represented (at a baseline level)	3/20	17	10	Agree = 53 Str agree = 6 *Total = 59*	[36, 78]
42. Each process relates to Maturity Level 2 (baseline processes)	*7/20*	*13*	10	Agree = 61.5% Str agree =15% *Total= 76.5%*	[54, 100]
45. All processes listed are at a similar level of abstraction	3/20	17	7	Agree = 35 Str agree = 6 *Total 41*	[22, 64]
55. The guidelines are at the same level of granularity.	2/20	18	16	Agree = 78 Str agree = 11 *Total = 89*	[67, 97]

Consistency between maturity levels appears strong with 94% support (Item 21). At this initial stage of development the maturity structure is modelled at a very high level, however the positive response suggests the R-CMM has a firm foundation.

Support for whether all key processes are represented at a baseline level is critical as the CI does not include the threshold value of 80% (Item 40). It is possible that the processes are not considered consistent with the baseline 'repeatable' process concept (Item 42). I need to ensure that I am guided by the SW-CMM concept and not the best practice literature that can introduce processes into the R-CMM that are not based on a logical order of implementation.

I listed the twenty candidate processes that according to my research qualified as key practices at a baseline level and asked the experts to rate them as 'Not Needed', 'Desirable', 'Essential' or 'Don't know'. Appendix N lists high level process definitions and how the experts rated each one. Points of interest are that each process is considered essential by one or more experts; only 3.75% of answers reflected a 'don't know' response (suggesting a reasonable level of understanding); 85.5% of answers reflected that the processes were either essential or desirable. Only 7.75% of answers suggested that the processes were not needed.

Additional comments from the experts revealed that a reason for using the 'not needed' category was because the process did not reflect the characteristics of a baseline process. One expert explained that the 'not needed' category was used against a process because it appears in a parallel project management key process area associated with other SPI models such as the CMMI (2001). The implication here being, that the process is needed, but not in a RE model.

Questions in this section highlighted a weakness in the R-CMM. An area in need of improvement is the consistency of process abstraction with the CI for Item 45 falling well below the 80% threshold. Yet, the more detailed guidelines that focus on one key process only appear to have a more consistent level of granularity with the CI including the 80% threshold (Item 55).

Criteria 4: Understandable

Questions in Table 34 test how easily the expert can interpret and understand the R-CMM.

Table 34: An indication of R-CMM meaning and comprehension

Questions	'No opinion' responses	No. of valid responses	Supportive responses: Observed	Supportive responses: %	Confidence Interval 95%: LL UL
24. How easy is it to understand the path from initial goal, to question, to final process?	0/20	20	17	(3) = 50 very = 35 *Total = 85*	[64, 95]
37. Each individual process is easy to understand (i.e. they are clearly defined and unambiguous)	3/20	17	5	Agree = 24 Str agree = 6 *Total = 30*	[13, 53]
47. Viewing requirements in 5 stages helps practitioners to understand when to implement each process	1/20	19	13	agree = 58 Str agree = 11 *Total = 69*	[46, 85]
80. How clear is this presentation of the model	0/20	20	17	(3) = 65 very = 20 *Total = 85*	[64, 95]

Clarity of presentation (Item 80) is given approximately 85% support. However with a fairly large CI, I do not infer that this support is necessarily representative of the population. Navigating from goals through to recommended processes receives a similar 85% support from the panel. This implies the goal focus is retained throughout the model description. The balance of agreement is that viewing RE activities in five stages helps practitioners to understand when to implement each process with 69% support. Yet, one expert stated that he would rather see the RE process in 'phases' rather than 'stages' as implied in the question.

The response to understanding individual processes (Item 37) is critical with just 30% of experts believing that definitions of processes in the R-CMM are clearly defined and unambiguous. This criticism could be due to the use of the 'SW-CMM' language where for example the process "Follow a written organisational policy for managing the system requirements allocated to the software project" is taken directly from the SW-CMM. We could therefore be compounding a recognised weakness in the SW-CMM. Yet one of the rules in my criteria states that language should be consistent with the SW-CMM. This may need to be revised.

One expert emphasises the need for clarity stating "The biggest problem with any of these models is interpretation, if the model can be interpreted differently it will be". This clearly is a major problem as I want the improvement effort to be repeatable, allowing organisations to view the state of their processes over time and between projects. If they interpret the processes differently, it is likely they are measuring different things. However with careful selection of assessors and participants as noted in the previous study, some of the problems associated with interpretation will be eliminated.

I am aware that the SW-CMM is sometimes viewed as having poorly defined processes and a lack of examples (for example, see (Lauesen and Vinter 2001) (Potter and Sakry 2001)). The R-CMM seems to suffer from similar criticism, where expressing each key RE process in terms that are universally understood is problematic. This is partly due to the lack of one industry standard or dictionary definition of terms that we can refer to Konrad and Shrum (2001). Taking processes from several sources has created a hybrid model that without further definitions appears ambiguous and vague. I am not surprised by the critical response to this item; I anticipated it by providing further definitions on a web page for my experts to refer to. However as these definitions were not included with the validation documentation I cannot tell if this form of support was indeed helpful.

What I can be confident about from these results however is that the definitions as they appear currently in the model are inadequate. They must either be more detailed at the level presented, or accompanied by definitions that are easy to access and understand. If the experts cannot understand the meaning behind the processes listed, it is also debatable whether they can accurately answer the questions related to their appropriateness as a key baseline process. More tests need to be undertaken in this area prior to proceeding with model development.

Criteria 5: Ease of use

Questions in Table 35 test expert perception of the level of ease with which the model might be implemented, i.e. how closely the model matches the practice it represents.

Table 35: An indication of R-CMM ease of use

Questions	'No opinion' responses	No. of valid responses	Supportive responses Observed	%	Confidence Interval 95% LL UL
22. How much previous knowledge of the SW-CMM do you think you need to be able to interpret this framework?	2/20	18	13	Fair knowl. = 56 No knowl. = 17 *Total = 72*	[49, 88]
47. Viewing requirements in these 5 phases is a reflection of how requirements are implemented in practice	0/20	20	13	Agree = 55 Str Agree = 10 *Total = 65*	[43, 82]
56. Dividing the requirements process into smaller activities in this way will help practitioners to implement the process	1/20	19	18	Agree = 58 Str Agree = 37 *Total = 95*	[75, 99]
76. How useful is it to take a process view of requirements to improve the overall requirements process?	2/20	18	18	(3) = 33 Very = 67 *Total = 100*	[82, 100]
84. How realistic is it to ask companies to look at their requirements process in this structured way?	2/20	18	10	(3) = 33 Very = 22 *Total = 55*	[34, 75]

There is very strong agreement that taking a process view of RE and dividing the RE process into smaller activities as given in the R-CMM will help practitioners implement the process (Items 76 and 56). Agreement suggests that the decomposition of processes from a high level description to a lower level prescription is helpful. However the perennial problem of bridging the gap between theory and practice is shown by 45% of my experts believing that it is unrealistic to expect companies to view their RE activities in this structured way (Item 84). One expert added, "... some agile development methods suggest much less RE activities (different RE activities for that matter); to what extent can your framework cope with a completely different view of RE...?"

The sentiment that the model structure is unhelpful is further confirmed by 35% of experts believing that the lifecycle view of requirements (i.e. requirements management, elicitation, negotiation, specification, verification) does not reflect software and system requirement practices (Item 47). Although this question does include my 80% acceptance threshold in the CI, it appears that the experts are reflecting the move away from the lifecycle view of the RE process.

A user does not require an in depth knowledge of the SW-CMM in order to interpret the R-CMM (Item 22). Although this question has the built-in assumption thatmymodel *is* similar to the SW-CMM, the response does indicate a high level of model independence that is likely to lead to a fast take of model concepts requiring minimal model-related training.

Criteria 6: Tailorable

Questions in Table 36 test how easily the R-CMM might be tailored to suit individual company needs.

The responses of 80% or above support in this category are all fairly positive, especially when considering adapting elements in the substantive framework in Item 25. Looking back to how the experts responded to the SW-CMM question on adaptability (Table 29, Item 15), the panel were not so enthusiastic with a lower support of 78% as opposed to the 95% support given to the R-CMM. The R-CMM potential for being adapted appears to be a relative strength of the design and a possible improvement on the SW-CMM. However, I appreciate that the model presented is at a fairly high level and the more detailed and more prescriptive the R-CMM becomes the less likely it is that the model can be tailored to suit all development environments.

Table 36: An indication of R-CMM tailorability

Questions	'No opinion' responses	No. of valid responses	Supportive responses: Observed	Supportive responses: %	Confidence interval 95%: LL	Confidence interval 95%: UL
25. How easy would it be to adapt this framework (e.g. {add|remove|amend} goals, questions and processes)?	1/20	19	18	(3) = 47.5 Very = 47.5 *Total = 95*	[75,	99]
66. How easy would it be to adapt this assessment method to meet individual company needs (e.g. measure different processes/use different measurement criteria)?	0/20	20	16	(3) = 70 Very = 10 *Total = 80*	[58,	92]
44. It would be possible to extend each process to create specific guidelines and prescriptions, i.e. convert process guidelines into practice.	1/20	19	18	Agree = 58 Str Agree = 37 *Total = 95*	[75,	99]
59. The activities [given in the guideline model component] are general and likely to apply to most companies.	1/20	19	15	Agree = 63.5 Str Agree = 16.5 *Total = 80*	[57,	91]

Criteria 7: Verifiable

I asked whether the questionnaire allowed the experts to give a fair assessment of the model. The questionnaire allowed 65% oftheexperts to give a fair opinion of the model, with 15% stating there was too much information. However, as there were only six missing responses in the entire questionnaire set of 1,700 responses I see little sign of questionnaire fatigue. Nearly all experts who felt that there was not enough detail to allow a fair assessment added further comments (with the exception of 1). However, I cannot be certain that these extra comments necessarily allowed the experts to explain precisely how they felt about the model. One expert suggested a different type of evaluation altogether, feeling that "applying the model to a project would allow me to evaluate the 'strengths and weaknesses' of the model more effectively". This would certainly be a more exacting test, and is considered for future work when I have addressed some of the more pressing issues that are raised in this study.

8.6.4 Overall impression

There is a near consensus that is would be helpful to continue work on the R-CMM to develop the high level process descriptions into more detailed guidelines. The experts' support for the R-CMM guidelines appear more pronounced when we compare it to their attitude towards the SW-CMM guidelines. The critical response in to the SW-CMM suggests a weakness (Table 29, Item 15), whereas the 96% support for the R-CMM guideline suggests a strength.

74% of experts believed that further development of the R-CMM would be useful to both the software industry and to the research community. However the experts were not unanimous and the following comments reflect this polarisation of attitudes:

"Hooking the RE process to the CMM is a great idea – many organisations have "bought into" the CMM process improvement initiatives and many organisations realise that poor RE is a source of myriad development problems. The association of these two ideas can go a long way toward improved RE processes".

Whereas another expert finds the connection with the SW-CMM unhelpful as reflected in this comment: "The problem with these checklist approaches is that they take no account of good process design... Tick box approaches offer an easy solution that ensures CMM compliance rather than good RE."

8.7 Summary of findings

The process of using an expert panel to validate the R-CMM has proved very helpful in highlighting some of the model's potential strengths and weaknesses. I believe that the involvement of such a high calibre panel that incorporates practitioners and researchers active in the field of RE and SPI adds weight and rigor to the results. The high response rate and the many additional comments and contributions made, suggest that the experts took the task seriously.

The range of responses elicited from this relatively small group formed a good basis for me to gauge how the R-CMM might be viewed in practice. It is a particularly worthwhile exercise as it provides an objective view on work that, otherwise, could easily become unrelated to the needs of the community. I therefore welcome the mix of opinions offered by this diverse group of experts.

8.7.1 R-CMM strengths and weaknesses

Despite some polarisation of views, agreement amongst the experts was relatively strong in the areas covered in Table 37.

Table 37: Summary of questionnaire results

R-CMM Strengths
The concept, 'to support the requirements process'
Retaining a SW-CMM concept (although some experts do not view this as a strength)
High level consistency of detail
A strong structure
Taking a process view of requirements
Decomposing activities
General adaptability/tailorability
Assessment component appears a good way to recognise requirements process weaknesses

R-CMM Weaknesses
The model appears incomplete (intentional at this level of abstraction)
Ambiguous process definitions (looking at high level definition alone is insufficient)
Unrealistic structured view (phases don't relate to requirements practices)
Structure may appear inflexible and detached
Inconsistent level of abstraction (a problem of combining multiple sources)
Missing key baseline processes
Wrong key baseline processes
The assessment component is not self-explanatory

These results have implications for future model development as discussed in the following section.

8.8 Reflecting the needs of the software industry

The R-CMM reflects the general needs of the software industry by attempting to provide a solution to recognised problems in the RE process. Results show that taking a process view of RE and creating a model that can be tailored to a company's individual needs is a worthwhile aim. However, the 5 phase lifecycle structure of the R-CMM does not necessarily reflect how requirements are implemented in practice. It is therefore questionable how helpful this dimension of the model is.

The model aims to represent RE process best practices. This element of the model proved the most contentious, with many experts believing that the R-CMM had either the wrong processes or missing processes. However when asked to rate the 'candidate' processes included in the model as being key to baseline processes, all processes were considered essential by some of the experts. Processes that were rated as 'not needed' represented a small percentage (7.75). However, the reason for rating some processes as not needed because they already appear in SPI models is a concern. If a process is rejected because it appears in other process improvement models, then many of the other processes should be rejected on the grounds that they appear in the SW-CMM. All processes included in the R-CMM are based on the needs of the practitioners in my previous empirical study that were using the SW-CMM to assist them with their improvement activities (Beecham *et al.* 2003d; Beecham *et al.* 2003f). RE processes therefore have been specifically sourced from the SW-CMM and the RE literature. Also, I did not intend the R-CMM to include *all*

RE processes, just those that are key to a majority of software development companies.

The mixture of responses from the experts suggests that it would be impossible to create a detailed requirements model that includes key RE processes that are relevant to all types of software development companies. The best I can expect is that there is a generic quality to the R-CMM that allows software companies to identify and adapt RE processes to meet their own needs.

Although the experts do not believe each process to be strictly 'requirements' related they did support combining organisational processes and technical processes in one model. I believe that the inclusion of organisational processes in the R-CMM ensures practitioners are given the freedom to concentrate on understanding and improving their technical RE processes. The technical processes are at a descriptive level that allows for creative adaptation.

One of the strengths of the R-CMM is its adherence to the SW-CMM. I can therefore expect that the prioritisation of process implementation offered by the SW-CMM is mirrored in the R-CMM. This logical decomposition of a complex system will help practitioners understand where and when their RE process needs improving. For example, the R-CMM would direct organisations to have a repeatable traceability process in place prior to exploring different methods for measuring requirements defects as it makes little sense to know the number of defects without knowing the cause of the defects.

The R-CMM provides an assessment method to help companies recognise where their current RE process needs strengthening. The experts were generally supportive of the assessment method with near consensus that it will highlight weaknesses and assist managers to prioritise their improvement activities. The weakness appeared to be that it will require further examples and definitions to be used effectively. Results therefore indicate that with limited guidance, the R-CMM is likely to help practitioners understand their current RE practices and where they need improving.

The R-CMM takes a top-down approach to improvement where each maturity level has its own goal. This goal is worked through from the SW-CMM maturity goal down to process level in a consistent and understandable way. Although some experts may view this structure as too rigid, the benefits offered include clear navigation between goals, requirements phases and processes. The R-CMM retains a goal focus throughout all model components. However, the R-CMM also takes a bottom-up approach by guiding practitioners to set their own specific goals, e.g. "Set realistic improvement goals to address problems in the RE process project" and "Monitor progress of the set requirements goals". The combination of a clear, strong structure and goal focussed processes should ensure that practitioners relate processes to goals.

Integration with other software development activities is achieved through adherence to the SW-CMM. Should an organisation want to incorporate their RE activities with other emerging improvement models, then this SW-CMM adherence will help their migration (Konrad and Shrum 2001). On a more detailed level the R-CMM key processes emphasise the need for all stakeholders to be involved in the RE process, where requirements are reviewed "to include software managers and other affected groups". However, the apparent ambiguity of process definitions in the R-CMM is a concern. If the processes that constitute the activities involved in the RE phase of development cannot be understood, there is going to be great difficulty in other phases in development tapping into them. If each user of the R-CMM views the processes differently there will be a loss of transparency and a likelihood of confusion. As this weakness is likely to impact most of my model objectives, strengthening process definitions is considered a priority in any future work. One expert highlights an apparent integration problem in CMM:

"The CMM approach still appears to rely on a sequential approach to acquiring and developing requirements. The reality in the current environment is that the elicitation, development and maintenance of requirements are very much ongoing activities throughout the implementation of a product. The CMM may and the RE model may support this, but it's not obvious from the presentation".

I appreciate that as I only presented the Level 2 processes in detail totheexperts, it may appear that there is an implicit order in this isolated group of activities. The apparent partitioned view of RE may appear inflexible. I need to re-think the presentation to reflect the on-going, cyclical nature of RE.

8.9 Conclusion

In this chapter I have shown how a group of experts validate the R-CMM. Their diverse response to the mix of questions in the questionnaire highlights some potential strengths and weakness of the R-CMM. The general attitude of the experts towards the model is supportive, with only two items being given less than a 50% supportive response throughout the entire questionnaire. However, I am also aware that having designed the survey instrument myself, there may be some bias in how I selected the questions.

The pattern of questionnaire responses suggests that the R-CMM is unlikely to appeal to all practitioners and researchers. However, the experts viewed the R-CMM as independent from the SW-CMM as their like or dislike of the SW-CMM did not follow through in their R-CMM related responses. It is helpful to the study that basing the R-CMM on a known framework does not appear to bias the results despite many experts having a firm opinion on the relative merits of the SW-CMM.

The results appear to show that the R-CMM does not reflect all kinds of RE development processes. This is shown in the experts' support for the high level framework that weakens as more detail is added to the model. Creating a model that is compatible with all software development needs is likely to be impossible when creating a detailed RE model. However, I believe that the strong framework that is well integrated with the SW-CMM the R-CMM still has potential as a basis for RE process improvement. Further work involves concentrating on the identified weaknesses to create a model the represents well-defined processes at a similar level of abstraction.

This validation study therefore serves as a guide to further development of the R-CMM. The research community can gain from this study as I explain my validation methodology in detail that allows for replication. The questionnaire results and attitudes of the experts towards RE and the SW-CMM as a SPI methodology are likely to be of interest to both the research community and the software industry.

PART FOUR

CONCLUSIONS

Chapter Nine: Conclusion

This chapter summarises the research programme. It explains how a model is developed to address RE problems in the software industry as detailed in my empirical studies and the literature. This R-CMM pulls together existing best practice to help practitioners gain a better understanding of the RE process. Understanding the RE process is the first part of the improvement strategy presented in this thesis. Once the process is understood and assessed, organisations will be in a better position to improve this complex phase of development. The work in this thesis is primarily motivated by findings in my empirical studies that highlighted the problems practitioners were experiencing with the RE process from within a SPI environment.

This concluding chapter includes a critique of the overall strategy and how, in hindsight, it might be improved. This chapter also reflects on the research methodology both in terms of its success and how it might be used in future research. Finally, suggestions are made as to how other research might evolve from the research presented in this thesis.

9.1 Summary of research findings

In this thesis I have shown how the R-CMM is designed, developed and validated. Results of the validation study indicate that the model adheres to SW-CMM characteristics where practitioners are guided towards developing a mature RE process capability.

This thesis examines my hypothesis that "A CMM-based RE process model can help to assess the maturity of the RE process." The resulting R-CMM can help to assess the maturity of the RE process through:

- Identifying RE processes
- Defining RE processes
- Prioritising RE processes

- Linking process improvement to maturity goals
- Assessing the strength and weaknesses of RE processes
- Involving key stakeholders

Results of a validation exercise indicate that the model:

- Gives a new focus to a software development area in need of support
- Has an architecture that is consistent with SW-CMM
- Lists key processes at a similar level of detail
- Has a strong/well-defined framework
- Decomposes RE processes
- Is adaptable and tailorable
- Has a good method for assessing RE process strengths and weaknesses

9.2 Overview of the work

The literature describes the RE process as a major impediment to software development. Weaknesses in requirements-related activities tend to have a damaging effect on the whole of software development. Empirical research conducted in this study re-affirms the RE process as a major problem for software practitioners. Companies are recognising their weaknesses in how they produce software and are using software process improvement models to support their development practices. The SW-CMM is the most applied SPI model and as such is found to support the RE process to an extent.

A review of the SW-CMM literature revealed the model to have many strengths and weaknesses. Independent studies show that successful implementation of the SW-CMM guidelines can lead to increased process capability. Assessing the capability of processes through a maturity structure helps organisations to understand where they are on the roadmap to producing reliable and predictable software.

In order to strengthen the RE process this research has explored the possibility of creating a specialised model of the RE process based on the SW-CMM architecture.

At a more detailed level of abstraction the research shows that baseline RE processes can be presented in a form that integrates with more mature RE processes as well as the whole of software development. To be truly useful, the research indicates that processes need to be at a guideline level of detail. However, to present guidelines in a list form is not helpful unless they are associated with goals and assessments. The R-CMM provides this framework.

Along with the strengths offered by a strong framework come weaknesses of appearing to be inflexible and rigid. The R-CMM is therefore inheriting some of the weaknesses inherent in the SW-CMM. However, I have shown that employing a Goal/Question/Process/Metric approach to RE process improvement, creates an interactive model where users adapt the framework to suit their own business needs. Through assessing the different processes, organisations can determine where their weak processes are, and determine where processes require strengthening in accordance with business needs. The assessment is independent of the formal SEI assessment and therefore gives organisations the freedom to apply it in conjunction with their own software process improvement programme.

9.3 Main findings drawn from the thesis

In the process of building the model the following findings are made:

9.3.1 RE process problems in context with the SW-CMM

This thesis examines the scale and shape of RE process problems and although I cannot generalise from my results, the results do offer insights into RE problems. The companies in the empirical study were are all using the SW-CMM as their method of software process improvement and represent a cross section of application areas and company sizes.

The results of a correspondence analysis of general software problems revealed RE problems as central to all practitioners. Within the problems recorded, there is a concentration of people and communication issues. Analysis by practitioner group revealed that although there was a general consensus as to the types of problems, each group had different needs. It is the developer who is most closely connected with RE process problems. And predictably, project managers are associated with project issues, and senior managers are associated with organisational matters. These results confirm the importance of considering the needs of the separate stakeholders in RE process improvement programmes.

9.3.2 Organisational and technical RE process problems

Problems and solutions are divided into organisational and technical processes to reflect the RE literature. Also, in managing the RE process it is likely that these classes of problems will require different treatment. For example organisational issues are likely to be under the control of management, while developers are more likely to be responsible for making improvements to the technical RE processes.

The study agrees with the literature in finding organisational problems to be more prevalent than technical problems in RE processes. Indeed, organisational problems are believed to amplify some technical process problems. The RE process problems highlighted in the literature, such as requirements growth and changing requirements, have less of an impact than expected on requirement process capability. Major impediments to producing a high quality requirements document appear related to poor communication channels between developers and customers, low staff retention, poor skills and a lack of training.

The SW-CMM seems to be supporting companies with their RE processes to an extent. Higher maturity companies tend to exhibit fewer RE problems. However, RE problems exhibited in higher maturity companies tend to stem from organisational issues rather than technical RE process issues. This lack of technical issues in the higher maturity groups indicates that technical processes are more resistant to 'damage' from organisational issues. This lack of technical problems could also be due to manager groups in high maturity companies exhibiting a greater understanding of the organisational issues. In agreement with the literature, this study finds that the capability of technical processes will only be enhanced when non-technical issues are improved.

9.4 Recommendations from the empirical study

- The study suggests that the SW-CMM is well constructed and provides further justification for using the improvement model to focus on RE problems voiced by practitioners.

- The differences in practitioner focus should be acknowledged if managers are going to achieve a universal 'buy-in' to SPI as there must be something in the improvement effort for everyone. Developers should be consulted and involved in process improvement initiatives as they are most aware of RE process inadequacies.

- Immature companies are especially susceptible to problems in the RE process. Given that 70% of software companies are said to remain at CMM level 1, the scale of RE problems across the industry could be very large.

- Organisational or non-technical RE process problems should be addressed in a planned way.

- Organisational process improvement activities should be integrated with technical improvements. These human-based problems are occurring in all levels of process maturity as characterised by the SW-CMM.

- It is critical to improve communication between developers and customers.

- Create an environment where practitioners want to stay for the long term.

- Conduct own assessment within the company to gain an understanding of individual process strengths and weaknesses. Don't rely on anecdote or individual case studies when prioritising RE process problems.

- Key practitioners should participate in the assessment from all staff groups involved in RE.

- Best practices can be learnt from higher level SW-CMM companies, as they exhibit fewer problems with RE processes.

The results of the empirical study into RE problems contributes to the increasing body of knowledge showing that improved RE process capability is related to improved organisational performance (El Emam and Birk 2000). Considering the criticality of requirements to project success, it appears that software companies are in need of further support with their RE process. A proposed solution is to create a CMM-based RE process model to assess the maturity of the RE process.

9.5 The contribution to knowledge

I adapt the SW-CMM 5 level maturity framework to focus on RE-related processes to create a unique solution to prevailing RE problems. The validated model presented in this thesis adds to the knowledge of model building methods, RE and software process improvement. No previous work presents the RE process in isolation in terms of individual process capability that is governed by SW-CMM maturity goals.

I have pulled together individual solutions under one framework using a process improvement approach already familiar to many practitioners. The R-CMM is a specialised, cohesive and comprehensive model that reflects RE key processes at incremental levels of capability. At this current stage of development I present a 5 level maturity framework with a detailed baseline level example of repeatable RE processes.

The unique R-CMM builds on the work of the SEI and the RE literature to create an outline model that combines technical RE processes together with supportive organisational processes. The R-CMM addresses known problems in the RE process and the SW-CMM model through a robust and transparent model building strategy rarely shown in software process improvement models and literature.

My contribution to knowledge is a model that represents a new process view of the requirements phase. Offering a full explanation of how the model is built, where there are strengths in the model, and where possible improvements can be made should enable other researchers to build on my work and continue towards seeking methods to improve the RE process.

This work has led to the publication of several papers as listed in Appendix O.

9.6 The R-CMM

The work in this thesis culminates in a series of model components that have been developed through following generic rules created at the start of development. As a starting point, practitioners are guided towards recognising baseline RE processes. The processes defined in the Level 2 R-CMM work together to produce a baseline structure for companies to consider within their software development activities. The clear definitions given in the guidelines will help with process implementation. Also, using the GQM approach will guide practitioners towards improving and managing RE through recognising the specific needs of the organisation. The R-CMM therefore guides users towards creating specific goals based on their business needs.

The model directs practitioners to examine their RE process in a systematic and detailed way. The R-CMM includes some SW-CMM best practices together with additional RE processes that are outside the scope of the SW-CMM. The study shows how processes included in the R-CMM that are not explicitly modelled in the SW-CMM are included on the basis of meeting the needs of practitioners in my empirical study, as well as taking best practices from the literature.

An expert panel validation of the R-CMM elicited a diverse range of responses to the mix of questions in the questionnaire. The general attitude of the experts towards the model was supportive, with only two items being given less than 50% support. Validation of the R-CMM shows the model has potential as an improvement tool where both practitioners and researchers in the field of SPI and RE are provoked into thinking and deliberating about the RE process.

The pattern of questionnaire responses suggests that the R-CMM is unlikely to appeal to all practitioners and researchers. However, the experts viewed the R-CMM as independent from the SW-CMM as their like or dislike of the SW-CMM did not follow through in their R-CMM related responses. It is helpful to this study that basing the R-CMM on a known framework does not appear to bias the results despite many experts having a firm opinion on the relative merits of the SW-CMM.

The results appear to show that the R-CMM does not reflect all kinds of RE development processes. This is shown in the experts' support for the high level framework that weakens as more detail is added to the model. Creating a model that is compatible with all software development needs is likely to be impossible when creating a detailed RE process model. However, I believe that the strong framework that is well integrated with the SW-CMM the R-CMM can be improved. Further work involves concentrating on the identified weaknesses to create a model that represents well-defined processes at a similar level of abstraction.

The validation study therefore serves as a guide to further development of the R-CMM. The research community can gain from this study as I explain my validation methodology in detail that allows for replication. The questionnaire results and attitudes of the experts towards RE and the SW-CMM as a SPI methodology are likely to be of interest to both the research community and the software industry.

9.7 Critique of methodology

This section presents the limitations identified in this research. It identifies issues that should be done differently if this research were to be repeated.

9.7.1 Qualitative data collection

Data collected through focus groups characterises abstract problems that have not been verified directly with the subjects. Theories were built from the focus group data through a grounded theory approach. The subjects were not directly asked about a) how they define the RE process; and b) what their main problems with RE processes are; and c) whether they believe RE to be the main cause of their software development problems. The nature of the study meant that I had no control over the questions asked as the data collection was undertaken prior to my involvement in the project. The design of the study would be improved if RE process problems were directly addressed, after the general software development problems had been covered. However as the questioning in the focus groups were necessarily structured and pre-planned, it is not always possible to pre-empt the pattern of responses.

An alternative way to collect data that does not depend on eliciting information from groups or individuals in an interview situation is to observe how the company operates. Ideally observations of how each company operated could be undertaken over a period of time to gain a further perspective on the depth and breadth of the RE process problems. Observing practitioners involved in RE would require a very large scale operation.

As the interpretation of problems in this thesis has not been directly verified, it could be that the classification of RE problems may differ from the practitioners' view. This certainly is the case in the literature where the term 'requirements' can range from a formal narrow understanding relating to a specific need of a system by a stakeholder to a broader definition meaning any requirements related process or model. A future study of RE process problems should build in specific open questions relating to how practitioners define the RE process, and what they believe are the inhibiting factors to building a high quality software requirements specification.

9.7.1.1 One to one interviews as opposed to focus group discussions

Even though over 200 software practitioners were involved in this research the data points used for analysis were less because focus groups were adopted in the data collection. One to one interviews would have resulted in a much larger data set where potentially higher frequencies of problems might result.

A further benefit of a one to one interview, is that each comment made can be attributed to one person. A problem of the focus group data is that this independence of data is compromised as a comment made by one person several times is given the same weighting as a comment made by several people once. Therefore an aggregate of problem frequencies collected from one-to-one interviews is more likely to reflect a more diverse sample. However, if the focus group is correctly managed, no one person should dominate the discussion (Greenbaum 1998).

One to one interviews are likely to be very time-consuming so would reduce the number of practitioners who could take part in the study Also, one to one interviews would isolate interviewees, rendering them less able to express the range of issues that they tend to do within the pseudo anonymity of a focus group (Baddoo 2002).

9.7.2 Quantitative data collection

9.7.2.1 The expert panel questionnaire survey

Using questionnaires to validate the SW-CMM placed an artificial limit on how much feedback the expert could report on perceptions about the model. Also, having designed the survey instrument myself, there may be some bias in my design of the questions. Ideally this form of validation should be conducted by a group or individual who has no investment in the results.

Although questionnaire surveys are often used to target a large sample, this was not the main incentive for using surveys in this study. In my case, I was more interested in the quality of the sample rather than the quantity. However, as the sample was relatively small this caused problems for the analysis of questionnaire results as many statistical methods are designed to be used with a larger sample (e.g. confidence intervals).

9.7.2.2 Using structured interviews as an alternative to the questionnaire survey

The use of structured interviews would have strengthened the validity of the model for three reasons. Firstly, the interviewee could interact with the interviewer and discuss any ambiguity in the question and give the interviewer the opportunity to confirm a level of understanding is reached. Secondly, the interviewer could confirm the extent to which the interviewee needed additional definitions and access to extra documentation to support the answers. Thirdly, releasing the participant from scale answers is likely to result in more diverse data that could lead to improvements in the model that cannot be anticipated in the questionnaire design.

The results of a structured interview would therefore result in a richer data collection, and would have proved valuable in guiding further model development, especially given the calibre of the participants. However the analysis of the results would have been more complex and it would have questionable validity in confirming levels of agreement and disagreement between participants. Also, as the format of the interview depends on the interviewee studying diagrams and model components, interaction with an interviewer might be a distraction. The interviewee might feel pressured to rush through, whereas, the pilot study showed that a quiet environment was required to allow the participant to focus on the documentation to which the questionnaire referred. Another disadvantage would be that this form of qualitative data collection would be more time consuming for both the interviewer and the interviewee.

Also, structured interviews would have reduced the data set size, as it would have been impossible to conduct interviews in Australia, Finland, USA as well as different geographical areas in the UK. The experts were chosen for their expertise, not their geographical location. The questionnaire survey made it possible to reach a diverse group of experts who happen to reside in different areas of the globe.

9.7.2.3 Sample size

Although the quantitative data collection made through the questionnaire had a high response rate, an increased sample would increase validity. Results of the parametric statistics used would become more representative of the total population if a larger number of subjects were used (Agresti and Coull 1998). Also, it would be better to know exactly why some of the subjects did not respond. Non-response can also skew the data.

9.8 Success and use of methodology in future applications

9.8.1 Focus groups

The use of focus groups in this research has been successful for generating large amounts of in-depth data. This method generated new ideas and further insights into problems practitioners were experiencing in their software development through rich anecdotal accounts. However, new ideas do not often come directly from the participants of focus groups - it is the responsibility of the researcher to interpret their comments (Greenbaum 1998).

Although focus group discussions were successful in uncovering specific problems the practitioners were experiencing in the software development, the quantity of data produced by this number of focus groups is vast, with over 1000 pages of text. It is therefore recommended in future that fewer groups are conducted with an emphasis on creating groups of comparable demographics. Although it is difficult to achieve in practice, it is recommended that the independent variables being explored have a consistent representation in the sample. For example there is a lack of validity in the comparison of patterns of behaviour between maturity levels when there are six companies at SW-CMM level 1 and one company at SW-CMM level 4. A more uniform representation would facilitate replication and comparison of the data.

9.8.2 Questionnaire surveys

The questionnaire survey data collection methodology gave me the freedom to select participants for their 'expertise' and not their location. The limit imposed by the design, that allowed only a few responses to each set question, eased the analysis process. The results from this small sample, though fairly flat, are very useful as an initial exploratory study into R-CMM strengths and weaknesses. However, in future applications, a larger sample of would allow for a greater confidence that the results are indeed representative of the population.

9.8.3 Expert panel

The process of using an expert panel to validate the R-CMM proved very helpful in highlighting some of the model's potential strengths and weaknesses. It is encouraging that experts in both academia and the software industry dedicated the time towards helping with this project. I would therefore recommend that other researchers consider this method as people in the industry are shown to be open to collaborating with new work, especially when they know there is a definite limit in the time required.

The involvement of such a high calibre panel adds weight and rigour to the results. The high response rate and the many additional comments and contributions made, suggest that the experts took the task seriously. The range of responses elicited from this relatively small group formed a good basis for me to gauge how the R-CMM might be viewed in practice. It is a particularly worthwhile exercise as it provides an objective view on work that, otherwise, could easily become unrelated to the needs of the community. The results from this validation exercise, though not conclusive, are invaluable in directing future work.

9.9 Future work

For real value to be gained from this research, follow-up work is essential as the results of the empirical investigation are not ends in themselves (Potts 1993). This study has been designed to enable future research to build on the results. The validation of the R-CMM and the development transparency provides a foundation for future development in the area of RE process improvement as shown in the following sub-sections.

9.9.1 RE process definitions

The R-CMM requires more precise and accessible process definitions as depending on high level definitions alone leads to ambiguity. How to improve the presentation of process definitions needs to be explored.

9.9.2 Conduct further research on baseline RE processes

Further research is required to uncover the key RE processes at a baseline level. The validation study and literature show that a consensus is unlikely to be gained. However, it would be worthwhile finding out whether key RE processes differ between companies of different sizes, application areas, process maturity and structure. Results from such research could be used to create more tailored models of RE processes. Results from this further research would also help to explain why some of the experts believed the R-CMM to have missing and incorrect key baseline processes.

9.9.3 Practical application trials

A way to collect direct practical feedback would be to ask a group of expert RE practitioners to apply the model. The model, with its assessment component, is at a stage where Level 2 candidate processes could be tested in the workplace. One of the experts on the panel suggested he would be more comfortable with this form of model verification.

9.9.4 Development of assessment tool

The assessment component of the R-CMM was well received by the experts. However there was some concern that more information and possibly training would be required in order for it to be used successfully. The information generated by such an exercise would be vast and administration is likely to be very time-consuming. To counter these problems a tool could be developed where stakeholders participate in an on-line questionnaire that covers all the key areas.

The assessment component allows users to immediately test their RE process. One of the experts on the panel suggested that applying this in practice would also serve to highlight weaknesses in the model design. Future work therefore would involve a group of expert practitioners applying the model in the workplace.

9.9.5 Development of guidelines

The detailed guideline component was a perceived strength of the model. To create a more complete model, all the processes listed at a high level could be developed into guidelines that relate more closely to how the processes could be implemented.

9.9.6 Development of process maturity levels 3, 4 and 5

RE process maturity levels 3, 4 and 5 have been covered in the R-CMM at a very high level of abstraction. In order to create a more complete model these levels should be given the same detail as the level 2 example. I envisage level 3 being a straightforward addition, as the SW-CMM concept can clearly translate to RE processes. The project level processes can be built on to produce company–wide definitions of processes. The SW-CMM level 4 characteristics also appear possible to adapt where the focus is on measuring RE processes to create a more controlled process. However, given the lack of data for level 5 companies, it is difficult to envisage how processes linked to the production of a requirements specification will undergo continual improvements. Developing a RE process model at this level will require collaboration with high maturity companies to uncover how they differ (and indeed whether they differ) from other less mature companies in their approach to RE.

THE END

References

Abdel-Hamid, T. K. and Madnick, S. 1989. Lessons Learned from Modeling the Dynamics of Software Development. Communications of the ACM 32 (12): 1426-1455.

Agresti, A. and Coull, B. 1998. Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions. The American Statistician 52 (2): 119-126.

Ahuja, S. 1999. Process Improvement in a Rapidly Changing Business and Technical Environment. Fourth Annual Software Engineering European Process Group Conference.

Altman, D. G. 1991. Practical Statistics for Medical Research. London, Chapman & Hall.

ami. 1992. A Quantitative Approach to Software Management. South Bank University.

Andreou, A. 2003. Promoting software quality through a human, social and organisational requirements elicitation process. Requirements Engineering 8: 85-101.

Antón, A. 1996. Goal-based requirements analysis. Proceedings of 1996 IEEE International Symposium on Requirements Engineering (RE'95): 136-144.

Arisholm, E. and Sjoberg, D. I. K. 2000. Towards a framework for empirical assessment of changeability decay. Journal of Systems and Software (53): 3-14.

Bach, J. 1994. The Immaturity of the CMM. American Programmer 7 (9): 13-18.

Bach, J. 1995. Enough about process: what we need are heroes. IEEE Software 12 (2): 96-98.

Bach, J. 1999. What Software Reality is Really About. IEEE Computer 32 (12): 148-149.

Bache, R. and Neil, M. 1995. Introducing metrics into industry: a perspective on GQM. Ed(s): N. Fenton, R. Whitty and Y. Iizuka. Software Quality Assurance and Measurement: A worldwide perspective International Thompson Computer Press, London.

Baddoo, N. 2002. Motivators and De-Motivators in Software Process Improvement: An Empirical Study. Academic PhD Thesis, University of Hertfordshire, Department.

Basili, V. R. 1995a. Applying the Goal/Question/Metric paradigm in the experience factory. Ed(s): N. Fenton, R. Whitty and Y. Iizuka. Software Quality Assurance and Measurement: A worldwide perspective International Thompson Computer Press, London.

Basili, V. R. 1995b. Measurement Frameworks. Ed(s): N. Fenton, R. Whitty and Y. Iizuka. Software Quality Assurance and Measurement: A Worldwide Perspective International Thomson Computer Press, London.

Basili, V. R. and Rombach, H. D. 1988. The Tame Project: Towards Improvement-oriented software environments. IEEE Transactions on Software Engineering 14 (6): 758-773.

Basili, V. R. and Turner, A. J. 1975. Iterative enhancement: a practical technique for software development. IEEE Transactions SE-1 (4): 390-396.

Baumert, J. 1994. Process Assessment with a Project Focus. IEEE Software 11 (2): 89-91.

Beecham, S., Hall, T., Britton, C., Cottee, M. and Rainer, A. 2003a. Using an expert panel to validate a requirements software process improvement model. IEEE Transactions on Software Engineering. In review.

Beecham, S., Hall, T. and Rainer, A. 2003b. Assessing the strength of a requirements process. Technical Report. TR 381. University of Hertfordshire.

Beecham, S., Hall, T. and Rainer, A. 2003c. Building a Requirements Process Improvement Model. Technical Report. TR 378. University of Hertfordshire.

Beecham, S., Hall, T. and Rainer, A. 2003d. Building a Requirements Process Improvement Model. Software Process: Improvement and Practice. In review.

Beecham, S., Hall, T. and Rainer, A. 2003e. Defining a Requirements Process Improvement Model. Technical Report. TR 379. University of Hertfordshire.

Beecham, S., Hall, T. and Rainer, A. 2003f. Defining a Requirements Process Improvement Model. Software Quality Journal. In review.

Beecham, S., Hall, T. and Rainer, A. 2003g. Software Process Improvement Problems in 12 Software Companies: An Empirical Analysis. Empirical Software Engineering 8 (1): 7-42.

Bell, T. E. and Thayer, T. A. 1976. Software Requirements: Are They Really a Problem? Proc. ICSE-2: 2nd International Conference on Software Engineering: 61-68.

Berry, D. and Lawrence, B. 1998. Requirements Engineering - Guest Editor's Introduction. IEEE Software: 26-30.

Berry, M. and Jeffery, R. 2000. An Instrument for Assessing Software Measurement Programs. Empirical Software Engineering 5: 183-200.

Billings, C. and Cliffton, J. 1994. Journey to a mature Software process. IBM Systems Journal 33 (1): 46-61.

Black, T. 1999. Doing Quantitative Research in the Social Sciences. London, Sage Publications Ltd.

Boehm, B. 1988. A spiral model of software development and enhancement. Computer 21 (5): 61-72.

Boehm, B. (2001). Using Win-Win for Requirements Negotiation: a mini tutorial. London,

Boehm, B., Bose, P., Horowitz, E. and Lee, M. 1994. Software Reqirements As Negotiated Win Conditions. Proceedings of the First International Conference on Requirements Engineering: 74-83.

Boehm, B. W. 1981. Software Engineering Economics. Englewood Cliffs, Prentice-Hall, Inc.

Britton, C. 1996. Software system development : a gentle introduction /2nd edition. London, McGraw-Hill.

Britton, C. 2000. Object-oriented system development : a gentle introduction. New York; London, McGraw-Hill.

Brodman, J. G. and Johnson, D. L. 1994. What Small Businesses and Small Organisations Say About the CMM. 16th International Conference on Software Engineering: 331-340.

Brooks, F. P. 1987. No Silver Bullet: Essence and accidents of Software Engineering. IEEE Computer 20 (4): 10-19.

Brooks Jr., F. P. 1987. No Silver Bullet: Essence and accidents of Software Engineering. IEEE Computer 20 (4): 10-19.

Bryman, A. 1996. Quantity and Quality in Social Research. London, Routledge.

Burnard, P. 1991. A method of analysing interview transcripts in qualitative research. Nurse Education Today (11): 461-466.

Burnstein, I., Suwannasart, T. and Carlson, C. 1996. Developing a Testing Maturity Model, Part II. CrossTalk The Journal of Defense Software Engineering.

Burton, R. 1621. Anatomy of Melancholy. Democritus to the Reader. 2003 The Clarendon Press/Oxford University Press edition (1989-94) is currently out of print. It comes in three volumes, with an Introduction by J.B.Bamborough, and was edited by Thomas C. Faulkner, Nicolas K. Kiessling, and Rhonda L. Blair (25.08.03) URL http://www.bartleby.com/100/.

Butler, K. and Lipke, W. 2000. Software Process Achievement At Tinker Air Force Base, Oklahoma. SEI, Carnegie Mellon University.

Carson, J. S. 1986. Convincing Users of Model's validity is Challenging Aspect of Modeller's Job. Industrial Engineering 18 (6): 74-85.

Cheng, B. and Weiss, D. 2000. Requirements engineering: integrating technology. IEEE Software 17 (3): 18-20.

Christie, A. M. 1999. Simulation in support of CMM-based process improvement. Journal of Systems and Software 46 (2-3): 107-112.

CMMI. 2001. Capability Maturity Model ®. Integration (CMMI SM), Version 1.1. December. Technical Report URL http://www.sei.cmu.edu/cmmi.

Coad, P. and Yourdon, E. 1990. Object Oriented Analysis. Ed(s): R. H. Thayer and M. Dorfman. System and Software Requirements Engineering IEEE Computer Society Press, Los Alamitos, CA.

Cohn, M. and Ford, D. 2003. Introducting an Agile Process to an Organization. IEEE Computer. June: 74-78.

Conradi, R. and Fuggetta, A. 2002. Improving Software Process Improvement. IEEE Software 19 (4): 92-99.

Conte, S., Dunsmore, H. and Shen, V. 1986. Software Engineering Metrics and Models. Menlo Park, California, The Benjamin/Commings Publishing Company, Inc.

Cottengim, D. 2002. Prerequisites for Success: Why Process Improvement Programs Fail. CrossTalk The Journal of Defense Software Engineering: 26-30.

Cramer, D. 1997. Basic Statistics for Social Research. London, Routledge.

Creswell, J. 1994. Research Design: Qualitative and Quantitative Approaches. Thousand Oaks, Sage.

Crosby, P. B. 1979. Quality is Free. New York, New American Library.

Crosby, P. B. 1986. Quality without Tears, McGraw-Hill.

Cugola, G. and Ghezzi, C. 1998. Software Processes: a Restrospective and a Path to the Future. Software Process - Improvement and Practice 4 (3): 101-123.

Curtis, B. 1994. A Mature View of the CMM. American Programmer 7 (9): 19-28.

Curtis, B. 2000. The Global Pursuit of Process Maturity. IEEE Software 17 (4): 76-78.

Curtis, B., Hefley, W. E., Miller, S. and Konrad, M. 1995. People Capability Maturity Model. CMU/SEI-95-MM-02. Software Engineering Institute, Carnegie Mellon University.

Curtis, B., Krasner, H. and Iscoe, N. 1988. A Field Study of the Software Design Process for Large systems. Communications of the ACM 31 (11): pp 1268-1287.

Daskalantonakis, M. K. 1994. Achieving Higher SEI Levels. IEEE Software 11 (4).

David, A. 2000. Models implementation: A state of the art. European Journal of operational Research 134: 459-480.

Davis, A. 1988. A Taxonomy for the Early Stages of the Software Development Life Cycle. Journal of Systems and Software: 297-311.

Davis, A. 1994. Giving Voice to Requirements Engineering. IEEE Software 11 (2): 12-16.

Davis, A. and Hickey, A. 2002. Requirements Researchers: Do we practice what we preach? Requirements Engineering 7: 107-111.

Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A. and Theofanos, M. 1993. Identifying and Measuring Quality in a Software Requirements Specification. In Proceedings of the First International Software Metrics Symposium: 141-152.

Davis, P. 1992. Generalizing Concepts of Verification, Validation and Accreditation for Military Simlation. R-4249-ACQ. Rand, Sant Monica, CA.

De Panfilis, S., Kitchenham, B. and Morfuni, N. 1997. Experiences introducing a measurement program. Information and Software Technology 39 (11): 745-754.

Deming, W. E. 1982a. Out of Crisis, Cambridge University Press.

Deming, W. E. 1982b. Quality, Productivity, and Competitive Position. Massachusetts Institute of Technology Center for Advanced Engineering Study.

Deming, W. E. 1986. Out of Crisis, MIT Press.

Denzin, N. and Lincoln, Y. 2000. Handbook of Qualitative Research, Second Edition. Thousand Oaks, Sage Publications, Inc.

Diaz, M. and Sligo, J. 1997. How Software Process Improvement Helped Motorola. IEEE Software (September/October): 75-81.

Dobson, J. and Strens, R. 1994. Organisational requirements definition for information technology systems. Proceedings of the First International Conference on Requirements Engineering (IEEE): 158-165.

Doftman, M. 1990. System and Software Requirements Engineering. Ed(s): M. Dorfman. System and Software Requirements Engineering IEEE Computer Society Press, Los Alamitos, CA.4-16.

Donzelli, P. and Iazeolla, G. 2001. A dynamic simulator of software processes to test process assumptions. Journal of Systems and Software 56 (1): 81-90.

Dorfman, M. 1990. System and Software Requirements Engineering. Ed(s): R. H. Thayer and M. Dorfman. System and Software Requirements Engineering IEEE Computer Society Press, Los Alamitos, CA.4-16.

Dorfman, M. and Thayer, R. H. 1997. Software Engineering. Los Alamitos, CA, IEEE Computer Society Press.

Drehmer, D. and Dekleva, S. 2001. A note on the evolution of software engineering practices. Journal of Systems and Software 57: 1-7.

Dubois, E. and Pohl, K. 2003. A Major Step toward a Mature Requirements Engineering Community: guest editors' introduction. IEEE Software 20 (1): 14-15.

Dunn, G. 1989. Design and Analysis of Reliability Studies. The statistical evaluation of measurement errors. New York, Oxford University Press.

Dutta, S. and van Wassenhove, L. 1997. An Empirical Study of Adoption Levels of Software Management Practices within European Firms. INSEAD Research Initiative in Software Excellence Working Paper.

Dutton, J. 1993. Commonsense Approach to Process Modeling. IEEE Software: 56-93.

Dybå, T. 2000. An Instrument For Measuring The Key Factors Of Success in Software Process Improvement. Empirical Software Engineering 5 (4): 357-390.

El Emam, K. 1999. Benchmarking Kappa: Interrater Agreement in Software Process Assessments. Empirical Software Engineering 4: 113-133.

El Emam, K. and Birk, A. 2000. Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability. IEEE Transactions on Software Engineering 26 (6): 541-566.

El Emam, K., Fusaro, P. and Smith, B. 1999. Success Factors and Barriers for Software Process Improvement. Ed(s): C. Tully and R. Messnarz. Better Software Practice For Business Benefit: Principles and Experience. IEEE Computer Society,, Los Alamitos.Chapter 16.

El Emam, K., Goldenson, D., McCurley, J. and Herbsleb, J. 1998. Success Or Failure? Modeling the Likelihood Of Software Process Improvement. International Software Engineering Research Network.

El Emam, K., Goldenson, D. R., McCurley, J. and Herbsleb, J. 2001. Modelling the Likelihood of Software Process Improvement: An Exploratory Study. Empirical Software Engineering (6): 207-229.

El Emam, K. and Jung, H. 2001. An empirical evaluation of the ISO/IEC 15504 assessment model. Journal of Systems and Software 59 (1): 23-41.

El Emam, K. and Madhavji, N. H. 1995a. A Field Study of Requirements Engineering Practices in Information Systems Development. Second IEEE International Symposium on Requirements Engineering: 68-80.

El Emam, K. and Madhavji, N. H. 1995b. The Reliability of Measuring Organizational Maturity, John Wiley & Sons.

El Emam, K. and Madhavji, N. H. 1996a. Does Organizational Maturity Improve Quality? IEEE Software 13 (5): 109-111.

El Emam, K. and Madhavji, N. H. 1996b. An Instrument for Measuring the Success of the Requirements Engineering Process in Information Systems Development. Empirical Software Engineering 1: 201-240.

El Emam, K., Quintin, S. and Madhavji, N. H. 1996. User Participation in the Requirements Engineering Process: an Empirical Study. Requirements Engineering Journal 1 (1): 4-26.

Eriksson, D. M. 2003. A framework for the constitution of modelling processes: A Proposition. European Journal of Operational Research 145: 202-215.

Everitt, B. 1992. The Analysis of Contingency Tables, 2nd Edition. London, Chapman & Hall.

Everitt, B. 1998. The Cambridge Dictionary of Statistics. Cambridge, UK, Cambridge University Press.

Fairley, R. and Rook, P. 1990. Risk Management for Software Development. Ed(s): R. H. Thayer and M. Dorfman. System and Software Requirements Engineering IEEE Computer Society Press, Los Alamitos, CA.387-400.

Faulk, S. 1990. Software Requirements: A Tutorial. Ed(s): R. H. Thayer and M. Dorfman. System and Software Requirements Engineering IEEE Computer Society Press, Los Alamitos, CA.

Fayad, M. E. 1997. Software Development Process: A Necessary Evil. Communications of the ACM 40 (9): 101-103.

Fenton, N. 2001. Viewpoint Article: Conducting and Presenting Empirical Software Engineering. Empirical Software Engineering 6 (3): 195-200.

Fenton, N., Pfleeger, S. L. and Glass, R. 1994. Science and Substance: A Challenge to Software Engineers. IEEE Software 11 (4): 88-95.

Ferraiolo, K. M. 2002. Considerations for Re-architecting the SEI CMM. 2002 (16.7.02) URL http://www.systemcorp.com/2002/downloads/frames/karen_frame.html.

Fink, A. 1995. How to Sample in Surveys. Thousand Oaks, Sage Publications.

Finkelstein, A. 1992. A software process immaturity model. ACM SIGSOFT Software Engineering Notes archive 17 (4): 22 - 23.

Fordham, R. G. 1999. Software Process Maturity: Is level five enough? 21st International Conference on Software Engineering: 611.

Fox, C. and Frakes, W. 1997. The Quality Approach: Is it Delivering? Communications of the ACM 40 (6): 25-29.

Frangos, S. 1998. Motivated humans for reliable software products. Microprocesors and Microsystems 21: 605-610.

Gass, S. 1983. Decision aiding models: Validation, Assessment, and Related Issues for Policy Analysis. Operations Research 31 (4).

Gause, D. C. and Weinberg, G. M. 1989. Exploring requirements: quality before design. New York, NY, Dorset House Publishing.

Gibbs, A. 1997. Focus Groups. Social Research Updata, No.19 URL http://www.soc.surrey.ac.uk/sru/SRU19.html.

Gilb, T. 1996. Level 6: why we can't get there from here. IEEE Software 13 (1): 97-98, 103.

Glaser, B. G. and Strauss, A. L. 1967. The Discovery Of Grounded Theory: Strategies For Qualitative Research., Aldine de Gruyter.

Glass, R. 1996. The Relationship between theory and practice in software engineering. Communications of the ACM 39 (11).

Glass, R. 1997. Revisiting the Industry/Academe Communication Chasm. Communications of the ACM 40 (6): 11-13.

Glass, R., Vessey, I. and Ramesh, V. 2002. Research in software engineering: an analysis of the literature. Information and Software Technology 44: 491-506.

Gomaa, H. and Scott, D. 1981. Prototyping as a tool in the specification of user requirements. in Fifth Int. Conf. on Software Engineering: 333-341.

Graham, I. 1998. Requirements engineering and rapid development : an object-oriented approach. Harlow, Addison-Wesley.

Greenacre, M. and Blasius, B. 1994. Correspondence Analysis in the Social Sciences Recent Developments and Applications. London, Academic Press Ltd.

Greenbaum, T. 1998. The Handbook for Focus Group Research. Second Edition. Thousand Island, CA, Sage Publications.

Greenhalgh, T. and Taylor, R. 1997. How to read a paper: Papers that go beyond numbers (qualitative research). BMJ 315: 740-743.

Gresse, C. and Briand, L. C. 1998. Requirements for the knowledge-based support of software engineering measurement plans. Knowledge-Based Systems 11 (2): 125-143.

Gross, D. and Yu, E. 2001. From Non-Functional Requirements to Design through Patterns. Requirements Engineering (6): 18-36.

Hackos, J. T. 1997. From Theory to Practice: Using the Information Process Maturity Model as a tool for Strategic Planning. Technical Communication (44): 369-381.

Hakim, C. 1987. Research Design: Strategies and Choices in the Design of Social Research. London, Routledge.

Hall, T., Beecham, S. and Rainer, A. 2002a. Requirements Problems in Twelve Companies: An Empirical Analysis. EASE.

Hall, T., Beecham, S. and Rainer, A. 2002b. Requirements Problems in Twelve Companies: An Empirical Analysis. IEE Proceedings for Software 149: 153-160.

Hall, T. and Fenton, N. 1996. Software quality programmes: a snapshot of theory versus reality. Software Quality Journal: 235-242.

Hall, T., Rainer, A. and Baddoo, N. 2002c. Implementing software process improvement: an empirical study. Software Process: Improvement and Practice 7 (1): 3-15.

Hall, T., Rainer, A., Baddoo, N. and Beecham, S. 2001. An Empirical Study of Maintenance Issues within Process Improvement Programmes in Software Industry. IEEE Conf Soft Maint: 422-430.

Hammer, T., Rosenbert, L., Huffman, L. and L, H. 1997. Measuring Requirements Testing. ICSE.

Harrison, R., Baddoo, N., Barry, E., Biffl, S., Parra, A., Winter, B. and Wuest, J. 1999. Directions and Methodologies for Empirical Software Engineering Research. Empirical Software Engineering 4 (4): 405-410.

Hather, R., Burd, E. and Boldyreff, C. 1996. A method for application management maturity assessment. Information and Software Technology 38 (11): 701-709.

Hayes, W. and Zubrow, D. 1995. Moving on Up: Data and Experience Doing CMM-Based Process Improvement. CMU/SEI-95-TR-008. Carnegie Mellon University.

Herbsleb, J. 1997. Software Quality and the Capability Maturity Model. Communications of the ACM 40 (6): 30-40.

Herbsleb, J. D., Carleton, A., Rozum, J., Siegel, J. and Zubrow, D. 1994. Benefits Of CMM-Based Software Process Improvement: Initial Results. CMU/SEI-94-TR-013. SEI Technical Report.

Herbsleb, J. D. and Goldenson, D. R. 1996. A systematic survey of CMM experience and results. 18th International Conference on Software Engineering: 323-330.

Hetzel, B. 1995. The sorry state of software practice measurement and evaluation. Ed(s): N. Fenton, R. Whitty and Y. Iizuka. Software Quality Assurance and Measurement: A worldwide perspective International Thompson Computer Press, London.97-109.

Hofmann, H. F. and Lehner, F. 2001. Requirements Engineering as a Success Factor in Software Projects. IEEE Software: 58-66.

Horvat, R. V., Rozman, I. and Gyorkos, J. 2000. Managing the Complexity of SPI in Small Companies. Software Process: Improvement and Practice 5 (1): 45-54.

Houdek, F. and Pohl, K. 2000. Analysing Requirements Engineering Processes: A Case Study. Proceedings of 11th International Workshop on Database and Expert Systems Applications: 983-987.

Humphrey, W., Snyder, T. and Willis, R. 1991. Software Process Improvement at Hughes Aircraft. IEEE Software 8 (4): 11-23.

Humphrey, W. S. 1988. Characterising the Software Process: A maturity framework. IEEE Software 5 (2): 73-79.

Humphrey, W. S. 1989. Managing the Software Process. Reading, Massachusetts, USA, Addison-Wesley Publishing Company, Inc.

Humphrey, W. S. 1997. Introduction to the Personal Software Process. Reading, M.A., Addison-Wesley.

Humphrey, W. S. 2000. Introduction to the Team Software Process. Reading, M.A., Addison-Wesley.

Humphrey, W. S. 2001. A Software Maturity Framework. Ed(s): R. Hunter and R. Thayer. Software Process Improvement IEEE Computer Society, Los Alamitos, California.53-63.

Humphrey, W. S. 2002. Three Process Perspectives: Organizations, Teams, and People. Annuls of Software Engineering 14: 39-72.

Humphrey, W. S., Kitson, D. H. and Kasse, C. 1989. The State of Software Engineering in Practice: A Preliminary Report. 11th International Conference on Software Engineering: pp. 277-288.

Ibanez, M. and Rempp, H. 1996. European User Survey Analysis. 2003 Site updated January 2002. (May, 2003) (available from URL http://www.esi.es).

IEEE. 1994. IEEE recommended practice for software requirements specifications, Std 830-1993. The Institute of Electrical and Electronics Engineers, Inc.: 1-32.

IEEE. 1997. Conference Proceedings. 3rd IEEE International Symposium on Requirements Engineering.

IEEE. 1998. IEEE Guide to Software Requirements Specification. IEEE Press.

IEEE. 1999. Conference Proceedings. 4th IEEE International Symposium on Requirements Engineering.

Ince, D. 1994. ISO9001 and Software Quality Assurance. London, McGraw-Hill Book Company.

ISO. 1999. Software Process Assessment - Part 5: An Assessment Model and Indicator Guidance. International Standards Organisation.

Jackson, M. 1995a. Problems and Requirements. Second IEEE International Symposium on Requirements Engineering: 2-8.

Jackson, M. 1995b. Software Requirements & Specifications: a lexicon of practice, principles and prejudices. Wokingham, England, Addison Wesley Publishing Company.

Jalote, P. 1997. An Integrated Approach to Software Engineering: Second Edition. New York.

Juran, J. M. 1981. Product Quality - A Prescription for the West. Management Review.

Kauppinen, M., Aaltio, T. and Kujala, S. 2002. Applying the Requirements Engineering Good Practice Guide for Process Improvement. Proceedings of the seventh European Conference on Software Quality (QC2002): pp 45-55.

Kitchenham, B., Pfleeger, S. L., McColl, B. and Eagan, S. 2002a. An empirical study of maintenance and development estimation accuracy. Journal of Systems and Software 64 (1): 57-77.

Kitchenham, B., Pfleeger, S. L., Pickard, L., Jones, P., Hoaglin, D., El Emam, K. and Rosenberg, J. 2002b. Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Transactions on Software Engineering 28 (8): 721-734.

Kitchenham, B. A., Hughes, R. T. and Linkman, S. G. 2001. Modeling Software Measurement Data. IEEE Transactions on Software Engineering 27 (9): 788-804.

Kitzinger, J. 1994. Focus Groups: Method or Madness? Ed(s): M. Boulton. Challenge and Innovation: Methodological Advances in Social Research on HIV/AIDS Taylor and Francis, London.

Knethen, A. v., Paech, B., Kiedaisch, F. and Houdek, F. 2002. Systematic Requirements Recycling through Abstraction and Traceability. Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02): 273-281.

Konrad, M. and Shrum, S. 2001. CMM Integration - Enterprise-Wide Process Improvement using the Next Generation of CMM's. Ed(s): R. Hunter and R. Thayer. Software Process Improvement IEEE Computer Society, Los Alamitos, California.

Kotonya, G. and Sommerville, I. 1998. Requirements Engineering : processes and techniques. Chichester, John Wiley & Sons.

Koubarakis, M. and Plexousakis, D. 2002. A formal framework for business process modelling and design. Information Systems 27: 299-319.

Kovitz, B. 2003. Hidden skills that support phased and agile requirements engineering. Requirements Engineering 8 (2): 135 - 141.

Krasner, H. 1997. Accumulating The Body Of Evidence For The Payoff Of Software Process Improvement. May 1999 URL http://www.utexas.edu/coe/sqi/archive/krasner/spi.pdf.

Krasner, H., Terrel, J., Linehan, A., Arnold, P. and H, W. 1992. Lessons Learned from a Software Process Modeling System. Communications of the ACM 35 (9).

Krasner, H. K. C., Pyles, J. S. and Wohlwend, H. S. 1994. A Case History of the Space Shuttle Onboard Systems Project. Technical Report. Technology Transfer # 94093551A-TR. SEMATECH Technology Transfer: 1-22.

Kratochvíl, M. 2002. UML xtra-light : how to specify your software requirements. New York, Cambridge University Press.

Krippendorff, K. 1980. Content Analysis An Introduction to Its Methodology. Beverly Hills, Sage Publications.

Landis, J. and Koch, G. 1977. Measurement of Observer Agreement for Categorical Data. Biometrics 33: 159-174.

Lauesen, S. and Vinter, O. 2001. Preventing Requirement Defects: An Experiment in Process Improvement. Requirements Engineering Journal 6 (1): 37-50.

Lavazza, L. and Valetto, G. 2000. Requirements-based estimation of change costs. Empirical Software Engineering 5 (3): 229-243.

Leffingwell, D. and Widrig, D. 2000. Managing software requirements: a unified approach. Reading, Massachusetts, Addison-Wesley.

Linberg, K. R. 1999. Software developer perceptions about software project failure: a case study. Journal of Systems and Software 49: 177-192.

Lindland, O. V., Sindre, G. and Solvberg, A. 1994. Understanding Quality in Conceptual Modeling. IEEE Software 11 (2): 42-49.

Loconsole, A. 2001. Measuring the requirements management key process area. Proc. of the ESCOM 2001: 67-76.

Loomes, M. J. and Jones, S. 1998. Requirements Engineering: A Perspective Through Theory-Building. Third International Conference on Requirements Engineering.

Loucopoulos, P. and Karakostas, V. 1995. System requirements engineering. London, UK, McGraw-Hill Book Company Europe.

Love, M. and Siddiqi, J. 2000. Assessing Software Process Improvement at South Yorkshire Police: A Case Study. Report. Sheffield Hallum University.

Lubars, M., Potts, C. and Richter, C. 1993. A Review of the State of the Practice in Requirements Modeling. In proceedings of the IEEE International Symposium on Requirements Engineering: 2-14.

Macaulay, L. 1996. Requirements for requirements engineering techniques. Proceedings of ICRE96, the Second International Conference on Requirements Engineering.

Macaulay, L. 1999. Seven-layer model of the role of the facilitator in Requirements Engineering. Requirements Engineering 4 (4): 38-59.

Maciszek, L. 2001. Requirements analysis and system design : developing information systems with UML. New York, Pearson Education.

Madhavji, N. H. 1991. The process cycle. Software Engineering Journal 6 (5): 234-242.

Marshall, C. and Rossman, G. 1989. Designing Qualitative Research. Newbury Park, Sage Publications.

Mashiko, Y. and Basili, V. R. 1997. Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement. Journal of Systems and Software 36 (1): 17-32.

Maxwell, A. 1975. Analysing Qualitative Data. London, Chapman and Hall.

MBNQA. 2003. The Malcolm Baldridge National Quality Award. 2003 (12.12.03) URL www.quality.nist.gov.

McConnel, S. 1996. Rapid Development. Microsoft Press 2.

McConnell, S. 2002. The Business of Software Improvement. IEEE Software: 5-7.

McFeeley, B. 1996. IDEAL: A User's Guide for Software Process Improvement. CMU/SEI-96-HB-001. Software Engineering Instutute, Carnegie Mellon University.

McNemar, Q. 1955. Psychological Statistics. New York, Wiley & Sons.

Mellis, W. 1998. Software Quality Management In Turbulent Times - Are There Alternatives To Process Oriented Software Quality Management? Software Quality Journal 7 (3/4): pp. 277-295.

Meyer, B. 1985. On Formalism in Specifications. IEEE Software 2 (1): 6-26.

Middleton, P. and McCollum, B. 2001. Management of process improvement by prescription. Journal of Systems and Software (57): 9-19.

Miller, R., Acton, C., Fullerton, D. and Maltby, J. 2002. SPSS for Social Scientists. Basingstoke, UK, Palgrave Macmillan.

Moitra, D. 1998. Managing Change for Software Process Improvement Initiatives: A Practical Experience-based Approach. Software Process: Improvement and Practice 4 (4): 199-207.

Morgan, D. L. 1997. Focus Groups As Qualitative Research (2nd ed.), Sage Publications.

Morgan, D. L. and Krueger, R. A. 1993. When To Use Focus Groups And Why, in D. L. Morgan (Ed) Successful Focus Groups: Advancing The State Of The Art. Newbury Park, Ca, Sage.

Moynihan, T. 2000. Coping with 'requirements-uncertainty': the theories-of-action of experienced IS/software project managers. Journal of Systems and Software (53): 99-109.

Mullins, L. J. 1993. Management and Organisational Behaviour. London, Pitman Publishing.

Neissink, F., Clerc, V. and Vliet, H. 2002. The IT Service Capability Maturity Model. October 2. URL http://www.itservicecmm.org.

Newton, R. R. and Rudestam, K. E. 1999. Your Statistical Consultant. Thousand Oaks, CA, Sage Publications.

Nguyen, L. and Swatman, P. 2003. Managing the requirements engineering process. Requirements Engineering 8: 55-68.

Ngwenyama, O. and Neilsen, P. 2003. Competing Values in Software Process Improvement: An Assumption Analysis of CMM From an Organizational Culture Perspective. IEEE Transactions on Engineering Management 50 (1): 100-112.

Nuseibeh, B. and Robertson, S. 1997. Making Requirements Measurable. Proceedings of 19th International Conference on Software Engineering (ICSE 97).

Olsson, T. and Runeson, P. 2001. V-GQM: A Feed-Back Approach to Validation of a GQM Study. Pro. of the Seventh International Software Metrics Symposium METRICS 2001: 236-245.

O'Neill, E., Johnson, P. and Johnson, H. 1997. Representations in co-operative software development: An initial framework. Proceedings of the International Workshop on Representations in Software Development: 33-42.

Oppenheim, A. 2001. Questionnaire Design, Interviewing and Attitude Measurement new edition. New York, Continuum.

Osterweil, L. 1986. Software Processes are software too. Proceedings of the Ninth International Conference of Software Engineering: 2-13.

Ott, L., Kinnula, A., Seaman, C. and Wohlin, C. 1999. The Role of Empirical Studies in Process Improvement. Empirical Software Engineering 4 (4): 381-386.

Panedo, M. H. and Shu, C. 1991. Acquiring experiences with the modelling and implementation of the project life-cycle process: the PMDB work. Software Engineering Journal 6 (5): 259-274.

Patel, N. V. 1999. The Spiral of Change Model for Coping with Changing and Ongoing Requirements. Requirements Engineering 4 (4): 77-84.

Paulish, D. and Carleton, A. 1994. Case Studies Of Software Process Improvement Measurement. Computer 27 (9): 50-57.

Paulk, M. C. 1993. Capability Maturity Model for Software. CMU/SEI-93-TR-24 Software Engineering Institute Version 1.1.

Paulk, M. C. 2001. Models and Standards for Software Process Assessment and Improvement. Ed(s): R. Hunter and R. Thayer. Software Process Improvement IEEE Computer Society, Los Alamitos, California.5-37.

Paulk, M. C. and Chrissis, M. B. 2000. The November 1999 High Maturity Workshop. Software Engineering Institute.

Paulk, M. C., Weber, C. V., Curtis, B. and Chrissis, M. B. 1995. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, Massachusetts, Addison Wesley Longman, Inc.

Penedo, M. and Shu, C. 1991. Acquiring experiences with the modelling and implementation of the project life-cycle process: the PMDB work. Software Engineering Journal: 259-274.

Perry, D. E., Staudenmayer, N. A. and Votta, L. G. 1994. People, Organizations and Process Improvement. IEEE Software 11 (4): 36-45.

Peters, J. and Pedrycz, W. 2000. Software Engineering : an engineering approach. New York, John Wiley & Sons.

Pfleeger, S. L. 1995. Maturity, Models, and Goals: How to Build a Metrics Plan. Journal of Systems and Software 31 (2): 143-155.

Pfleeger, S. L. 1999. Albert Einstein and Empirical Software Engineering. IEEE Computer 32 (10): 32-37.

Pfleeger, S. L. and Rombach, H. D. 1994. Measurement based Process Improvement. IEEE Software 11 (4): 9-11.

Pidd, M. 1999. Just Modeling Through: A Rough Guide to Modeling. INTERFACES 29 (2): 118-132.

Pinheiro, F. 2003. Requirements Honesty. Requirements Engineering 8 (3): 183-192.

Pitterman, B. 2000. Telcordia Technologies: The Journey To High Maturity. IEEE Software 17 (4): 89-96.

Potter, N. and Sakry, M. 2001. Practical CMM. 2002 March 2001. (9.27.02) URL http://www.sdmagazine.com/print/documentID=11079.

Potter, W. J. 1996. An Analysis of Thinking and Research about Qualitative Methods. Mahwah, New Jersey, Lawrence Erlbaum Assocates.

Potts, C. 1993. Software Engineering Research Revisited. IEEE Software 10 (5): 19 - 28.

Potts, C. 1995. Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-shelf Software. Proceedings of 1995 IEEE International Symposium on Requirements Engineering (RE'95): 128-130.

Potts, C. 1997. Requirements Models in Context. Third IEEE International Symposium on Requirements Engineering: 102-104.

Pressman, R. 2001. Software Engineering 5th Edition, McGraw Hill.

Procaccino, J. D., M, V. J., Overmyer, S. P. and Darter, M. E. 2002. Case Study: factors for early prediction of software development success. Information and Software Technology 22: 53-62.

QFD. 2003. Quality Function Deployment model. (12.12.03) URL www.qfdi.org.

Rainer, A. and Hall, T. 2002. A quantitative and qualitative analysis of factors affecting SPI. Journal of Systems and Software.

Rainer, A. and Hall, T. 2003. A quantitative and qualitative analysis of factors affecting software processes. The Journal of Systems and Software 66 (1): 7-21.

Reifer, D. J. 2000. The CMMI: it's formidable. Journal of Systems and Software 50: 97-98.

Rifkin, S. 2001. Why Software Process Innovations Are Not Adopted. IEEE Software.

Robinson, S. 1997. Simulation Model Verification and Validation: Increasing the User's Confidence. Proceedings of the 1997 Winter Simulation Conference.

Rodeghier, M. 1996. Surveys with Confidence: A practical guide to survey research using SPSS. Chicago, USA, SPSS Inc.

Rogoway, P. 1998. How to Reap the Business Benefit from SPI: Adding SPICE while preserving the CMM (Motorola). 2003 Rated Newspaper Supported by EU Project SELECT (16.2.03) URL http://www.iscn.at/select_newspaper/assessments/motorola.html.

Rosqvist, T., Koskelo, M. and Harju, H. 2003. Software Quality Evaluation Based on Expert Judgement. Software Quality Journal 11 (1): 39-55.

Rossi, S. 1999. Moving Towards Modelling Oriented Software Process Engineering: A Shift from Descriptive to Prescriptive Process Modelling. International Conference on Product Focused Software Process Improvement: 508-522.

Royce, W. 1970. Managing the Development of Large Software Systems. IEEE proceedings: 1-9.

Royce, W. 1987. Managing the development of large software systems. Proc. 9th Int Conf on Software Engereering.

Rule, G. P. 2001. Using measures to understand requirements. European Software Conference on Metrics (ESCOM-2001) Conference Proceedings.

Sargent, R. 2000. Verification, Validation, and Accreditation of Simulation Models. Winter Simulation Conference: 50-59.

Sawyer, P., Sommerville, I. and Viller, S. 1997. Requirements Process Improvement Through the Phased Introduction of Good Practice. Software Process Improvement and Practice 3 (1).

Scharer, L. 1990. Pinpointing Requirements. Ed(s): R. H. Thayer and M. Dorfman. System and Software Requirements Engineering: IEEE Computer Society Press Tutorial IEEE Computer Society Press, Los Alamitos, CA.

Seaman, C. 1999. Qualitative Methods in Empirical Studies of Software Engineering. IEEE Transactions on Software Engineering 25 (4): 557-572.

SE-CMM. 2003. (12.12.03) URL http://www.sei.cmu.edu/cmm/se-cmm.html.

SEI. 1996. A Description of the Systems Engineering Capability Maturity Model Appraisal Method Version 1.1. Report. SECMM-94-06CMU/SEI-96-HB-004 v1.1. Software Engineering Institute, Carnegie Mellon University.

SEI. 2002a. Capability Maturity Model ® Integration (CMMI SM), Version 1.1 CMMI SM for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1)Staged Representation. Technical Report. CMU/SEI-2002-TR-012. ESC-TR-2002-012. Carnegie-Mellon Univ: 1-729.

SEI. 2002b. Capability Maturity Model ® Integration (CMMI SM), Version 1.1. CMMI SM for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1). Continuous Representation. Technical Report. CMU/SEI-2002-TR-01. ESC-TR-2002-011. Carnegie-Mellon Univ: 1-724.

SEI. 2003a. Capability Maturity Model® for Software (SW-CMM®). 2003 (September) URL http://www.sei.cmu.edu/cmm/.

SEI. 2003b. Process maturity profile of the Software Community. 2003 (June 2003) URLs http://www.sei.cmu.edu/sema/pdf/SW-CMM/2003apr.pdf and http://www.sei.cmu.edu/sema/profile_about.html.

SEI. 2003c. Standard CMMI® Appraisal Method for Process Improvement (SCAMPISM), Version 1.1: Method Definition Document. 2003 (Sept 2003) URL http://www.sei.cmu.edu/publications/documents/01.reports/01hb001.html.

Sharp, H. 1994. Knowledge Acquisition or Requirements Analysis? Proceedings The Second International Conference on Expert Systems for Development.

Shaw, I. 1999. Qualitative Evaluation. London, Sage Publications.

Shaw, M. 2002. What makes good research in software engineering? International Journal on Software Tools for Technology Transfer (STTT) 4 (1): 1 - 7.

Sheard, S. 2001. The Frameworks Quagmire. Ed(s): R. Hunter and R. Thayer. Software Process Improvement IEEE Computer Society, Los Alamitos, California.39-47.

Shere, K. D. 1988. Software Engineering and Management. Englewood Cliffs, N.J., Prentice Hall, Inc.

Shewhart, W. 1931. Economic Control of Quality of Manufactured Product. New York, Van Nostrand Company.

Shipman, M. 1997. The Limitations of Social Research: fourth edition. New York, Addison Wesley Longman Limited.

Siegel, S. and Castellan, N. J. 1988. Nonparametric Statistics for the Behavioral Sciences. Second Edition. New York, McGraw-Hill International Editions.

Silverman, D. 1993. Interpreting Qualitative Data: Methods for Analysing Talk, Text and Interaction. London, Sage Publications.

Six Sigma. 2003. 2003 (12.12.03) URL www.6-sigma.com.

Smith, R. C. 1998. Using a Quality Model Framework to Strengthen the Requirements Bridge. International Conference on Requirements Engineering (ICRE '98): 118-125.

Solingen, R. v. and Berghout, E. 1999. The Goal/Question/Metric Method: a practical guide for quality improvement of software development. Maidenhead, UK, McGraw-Hill Publishing Company.

Solingen, v. R. and Berghout, E. 2001. Integrating Goal-Oriented Measurement in Industrial Software Engineering: Industrial Experiences with and Additions to the Goal/Question/Metric Method (GQM). Pro. of the Seventh International Software Metrics Symposium METRICS 2001: 246-259.

Sommerville, I. 2001. Software Engineering. Wokingham, Addison-Wesley Publishing Company.

Sommerville, I. and Sawyer, P. 1997. Requirements Engineering A good practice guide. Chichester, John Wiley & Sons Ltd.

SPICE. 2003. Software Process Improvement and Capability dEtermination. (12.12.03) URL www-sqi.cit.gu.edu.au/spice.

Sprent, P. 1993. Applied Nonparametric Statistical Methods (2nd ed). London, UK, Chapman & Hall.

SPSS. 1996. Surveys with Confidence: A practical guide to survey research using SPSS. Inc.

SPSS. 1999. SPSS For Windows:System Reference Guide. Inc.

SPSS (2001). SPSS for Windows. Release 11.0. Standard Version,

Srinivasan, A. and Te'eni, D. 1995. Modeling as Constrained Problem Solving: An Empirical Study of the Data Modeling Process. Management Science 41 (3): 419-434.

Standish Group. 1995. Standish Group Chaos Report. URL http://www.scs.carleton.ca/~beau/PM/Standish-Report.html.

Stelzer, D. and Mellis, W. 1998. Success Factors of Organizational Change in Software Process Improvement. Software Process - Improvement and Practice 4 (4): 227-250.

Strok, D. 2003. Face to face with the Experts. IEEE Software: 92-93.

STSC. 2003. Mapping CMMI-SE/SW v1.1 to SW-CMM v1.1. 2003 (12/9/03) URL http://www.stsc.hill.af.mil/consulting/cmmi/documents.html.

Sutcliffe, A. G., Economou, A. and Markis, P. 1999. Tracing Requirements Errors to Problems in the Requirements Engineering Process. Requirements Engineering (4): 132-151.

Sutcliffe, A. G., Maiden, N. and Bright, B. 1997. An evaluation framework for representations in requirements engineering. Proceedings of the International Workshop on Representations in Software Development: 16-32.

Tesch, R. 1990. Qualitative Research: Analysis Types & Software Tools. New York, The Falmer Press.

Thayer, R. H. and Dorfman, M. 1990. System and Software Requirements Engineering. Los Alamitos, CA, IEEE Computer Society Press.

Thomas, M. and McGarry, F. 1994. Top down vs bottom up process improvement. IEEE Software 11 (4): 12-13.

Thomson, H. E. and Mayhew, P. 1997. Approaches to Software Process Improvement. Software Process - Improvement and Practice 3: 3-17.

Tichy, W. 1998. Should computer scientists experiment more? IEEE Computer 31 (5): 32-40.

Tichy, W., Lukowicz, P., Prechelt, L. and Heinz, E. 1995. Experimental evaluation in computer science. Journal of Systems Software 28 (1): 9-18.

van Lamsweerde, A. 2000. Requirements Engineering in the Year 00: A Research Perspective. 22nd International Conference on Software Engineering (ICSE).

Vogt, W. P. 1999. Dictionary of Statistics and Methodology. 2nd Edition, Sage publications.

Walker, R., Briand, L., Notkin, D., Seaman, C. and Tichy, W. 2003. Panel: Empirical Validation - What, Why, When, and How. Proceedings of the 25th International Conference on Software Engineering (ICSE '03).

Watkins, R. and Neal, M. 1994. Why and How of Requirements Tracing. IEEE Software (July): 104-106.

Webb, C. 1999. Analysing qualitative data: computerized and other approaches. Journal of advanced Nursing 29 (2): 323-330.

Weinberg, G. M. 1998. The psychology of computer programming. New York, NY, Dorset House Publishing.

Weller, S. and Ro

Wells, M. and Harrison, R. 2000. Multidisciplinary Solutions to Multidisciplinary Problems. Proceedings of the ICSE 2000 Workshop 'Beg, Borrow or Steal: Using Multidisciplinary Approaches in Empirical Software Engineering Research.

Weller, S. C. and Romney, A. K. 1990. Metric Scaling: Correspondence Analysis, Sage Publications Inc. Newbury Park, California.

Wiegers, K. 1998a. Molding the CMM to Your Organization. Software Development 6 (5).

Wiegers, K. 1998b. Read My Lips: No New Models! IEEE Software 15 (5).

Willis, R. R., Rova, R. M., Scott, M. D., Johnson, M. I., Ryskowski, J. F., Moon, J. A., Shumate, K. C. and Winfield, T. O. 1998. Hughes Aircraft's Widespread Deployment Of A Continuously Improving Software Process. Technical

Report. CMU/SEI-98-TR-006. Software Engineering Institute, Carnegie Mellon University.

Wohlin, C. 2003. Empirical Research Methods in Software Engineering. Lecture Notes in Computer Science 2765 / 2003: 7 - 23.

Wohlwend, H. and Rosenbaums, S. 1994. Schlumberger's Software Improvement Program. IEEE Transactions on Software Engineering 20 (11): 833-839.

Wood, J. and Silver, D. 1995. Joint Application Development. New York, John Wiley &Sons Inc.

Yamamura, G. 1999. Head to Head: Process Improvement Satisfies Employees. IEEE Software: 83-85.

Yu, E. S. K. and Mylopoulos, J. 1997. Modelling Organizational Issues for Enterprise Integration. International Conference on Enterprise Integration and Modelling Technology.

Zelkowitz, M., D, W. and Binkley, D. 1998. Culture Conflicts in Software Engineering Technology Transfer. Technical Report No.2. University of Maryland, College Park, Md.

APPENDICES

APPENDICES

Page No.

Appendix A: Glossary of Terms and Acronyms

Acronym	Meaning in full (For further definitions see glossary table)
CA	Correspondence Analysis
CI	Confidence Interval
CMM	Capability Maturity Model
CMMI®	Capability Maturity Model® Integration
EPSRC	Engineering and Physical Sciences Research Council
KPA	Key Process Area
PPP Project	'Managing Practitioner impact on Processes and Products' (PPP) project. A project funded by the UK's Engineering and Physical Science Research Council under grant number EPSRC GRL91962.
RE	Requirements Engineering
RM	Requirements Management
R-CMM	Requirements Capability Maturity Model
SCE	Software Capability Evaluation
SEI	Software Engineering Institute
SPI	Software Process Improvement
SRS	Software Requirements Specification
SW-CMM	Software Capability Maturity Model
GQM	Goal Question Metric
MIS	Management Information Systems

TERM	RELATED TERM AND ACRONYM	Definition of term that applies to this Thesis
Abstraction		The principle of ignoring those aspects of a subject that are not relevant to the current purpose in order to concentrate more fully on those that are (Coad and Yourdon, 1990).
Agile methods	E.g. "extreme programming"	Agile methods emphasise the non-technical aspect of developing software where software development is viewed as a highly social activity. Agile approaches are related to the 'inspect' and adapt' engineering approach where cycles and feedback loops are short (Cohn and Ford 2003).
Bespoke	Customised product	Systems which are commissioned by a particular customer. E.g. control systems for electronic devices, systems written to support a particular business process and air traffic control systems (Sommerville 2001)
Best Practice		A [proven] tactic or method chosen to perform a particular task and/or to meet a particular objective (Dooley et al. 2001).
Chi-Square	Used with Crosstabs in SPSS	Tests the hypothesis that the row and column variables are independent, without indicating strength or direction of the relationship. Pearson chi-square, likelihood-ratio chi-square, and linear-by-linear association chi-square are displayed. Fisher's exact test and Yates' corrected chi-square are computed for 2x2 tables.
Commercial off the shelf System	COTS or generic	There are many different definitions of COTS as listed in (Morisio and Torchiano 2002). The SEI definition is: A COTS product is sold, leased, or licensed to the general public; offered by a vendor trying to profit from it; supported and evolved by the vendor, who retains the intellectual property rights; available in multiple, identical copies; and used without source code modification.
Computer science		Computer science is concerned with the theories and methods which underlie computers and software systems (whereas software engineering is concerned with practical problems of producing software). (Sommerville 2001)
Contingency Table	Cross-tabs (in SPSS)	Data classified with respect to two or more variables (Everitt 1977). SPSS v 11.0 (2001) refers to Crosstabs as a procedure that forms two-way and multiway tables. The structure of the table and whether categories are ordered determine what test or measure to use. Crosstabs' statistics and measures of association are computed for two-way tables only. If you specify a row, a column, and a layer factor (control variable), the Crosstabs procedure forms one panel of associated statistics and measures for each value of the layer factor (or a combination of values for two or more control variables).
Correspondence Analysis	Descriptive Statistics	Data in a two-dimensional contingency are represented in the same geometrical space allowing examination of relations among row or column variables and between row and column variables (Weller and Romney 1990).
Culture	Organisational	"that's the way we do things around here" (Paulk, 1997, page 10)

Customer		The person, or persons who pay for the product and usually (but not necessarily) decide the requirements. In the context of this and the IEEE (1998) recommended practice the customer and the supplier may be members of the same organization. The individual, group, organisation that commissions the development of the system (Loucopoulos and Karakostas 1995).
Encapsulation		(see 'Information Hiding')
Engineering		"Engineering is the use of principles to find designs that will meet multiple competing objectives, within limited resources and other constraints, under conditions of uncertainty" (Gilb 1996).
Engineering Requirements	(see also requirements engineering)	Deals with activities which attempt to understand the exact needs of the users of the software intensive system and to translate such needs into precise and unambiguous statements which will subsequently be used in the development of the system (established as a separate field of investigation and practice in mid 1970s) (Loucopoulos and Karakostas 1995)
Framework		An essential supporting or underlying structure (Concise Oxford Dictionary 2001)
Goal Question Metric	GQM	A paradigm proposed by (Basili and Rombach 1988) that is used to help decide what measurements should be taken and how they should be used (Sommerville 2001).
Information Hiding	Encapsulation	A principle, used when developing an overall program structure, that each component of a program should encapsulate or hide a single design decision. The interface to each module is defined in such a way as to reveal as little as possible about its inner workings (Coad and Yourdon, 1990)
Inheritance		Properties or characteristics received from an ancestor (Coad and Yourdon, 1990)
Item	In Questionnaire	The term 'item' is used to mean the question and all its associated results; i.e. the row of results (El Emam and Jung 2001)
Life cycle	Software	The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and sometimes, retirement phase. These phases may overlap or be performed iteratively (IEEE 1999)
Life cycle	System	The period of time that begins when a system is conceived and ends when the system is no longer available for use (IEEE 1999)
Mann-Whitney U	Statistic method	A non-parametric test used to compare the responses of two independent groups.
McNemar Test	Statistic Method	A non-parametric test that compares binary response patterns between two matched conditions
Model		A simplified description.. of a system or process .. to assist predictions (Concise Oxford Dictionary 2001)
Multivariate	Statistics term	"an assortment of statistical methods that have been developed to handle situations in which multiple variables or measures are involved. Any analysis of more than two variables or measures can loosely be considered a multivariate statistical analysis"(Marcoulides and Hershberger 1997). "Having or involving a number of independent mathematical or statistical variables" (Webster's Dictionary: http://www.m-w.com/cgi-bin/dictionary)
Null hypothesis		The 'no difference' or 'no association' hypothesis to be tested (usually by means of a significance test) against an alternative hypothesis that postulates non-zero difference of association (Everitt 1998)
Normative		Relating to or deriving from a standard or norm (Oxford Dictionary, 2001)
Organisation	Methods of	According to Coad and Yourdon (1990) three methods pervade people's thinking: (1) the differentiation of experiences into particular objects and their attributes; (2) the distinction between whole objects and the component parts; and (3) the formation of and the distinction between different classes of objects. Taken from "Classification Theory", Encyclopedia Britannica.
Organisational RE Process Problems	In PPP project Project management problems	A class of factors internal to the development organisation that indirectly influence the production of the RE Specification that is often the responsibility of management, to include: company culture; developer communication; resources; skills; staff retention; training; user communication.
p-value		The *p* value shows the probability that an observed result (or result of a statistical test) is due to chance rather than to participation in a program (Cramer 1997).
Paradigm	Worldview	A basic set of beliefs or assumptions that guide qualitative researchers inquiries (Cresswell 1998)
Population	As used in statistics	A generic term denoting any well defined class of people or things (Everitt, 1977)
PPP Project		'Managing Practitioner impact on Processes and Products' (PPP) project. A project funded by the UK's Engineering and Physical Science Research Council under grant number EPSRC GRL91962.
Practitioner		People actively involved in producing software, to include developers, project managers and senior managers.
Practitioner Communication		Communication between staff groups within the Company. E.g. Marketing discussing customer needs and agreements with Software Group, or Requirements Engineers communicating feasibility of design with Software group.

Process		A collection of activities with entity flows among them (Yu and Mylopoulos 1997) or "particular method of doing something, generally involving a number of steps or operations." Webster's Dictionary in Fayad, 1997.
Process	Maturity	The degree, to which a process is defined, managed, measured, and continuously improved (Dooley et al. 2001).
Process	Tailoring or Customising	"for any process model to be effective in the specific project in hand, there is a need to customise the model according to the project goals. This may be achieved by characterising various aspects of the project (e.g. resource constraints); setting up project goals; assessing how these goals are supported by the adopted process model, tailoring the process model to suit project goals; using the tailored process model in the project; assessing and fine-tuning the model on an on-going basis. The customisation process would be simplified considerably if process models were organised hierarchically, leading from generic models at the top of the hierarchy to specific models at the bottom."(Madhavji 1991) (the cmm does this to an extent).
Qualitative Data		"When the population is classified into several categories we may then 'count' the number of individuals in each category. These 'counts' or frequencies are qualitative data." (Everitt 1977)
Quantitative Data		Data obtained from measurement of continuous variables such as height, temperature, etc. (Everitt, 1977)
Requirement	Or, set of Requirements	A feature or behaviour of the system that is desired by one or more stakeholders (Britton 2000). A condition or capability needed by a user to solve a problem or achieve an objective. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. A documented representation of a condition or capability as in (1) and (2) (IEEE 1999)
Requirements	Allocated (as in CMM)	The agreement with the customer of the requirements for the software project (Davis et al. 1993)
Requirements	Analysis	The process of studying user needs to arrive at a definition of system, hardware, or software requirements. The process of studying and refining system, hardware, or software requirements (IEEE 1999)
Requirements	Errors	2 classes according to (Davis et al. 1993) Knowledge errors: caused by not knowing what the true requirements are 2. Specification errors: caused by not knowing how to adequately specify requirements
Requirements	Defects	2 classes according to (Lauesen and Vinter 2001) 1. Requirements Defects: We have a requirement defect if the product works as intended by the programmers, but doesn't match the surroundings. One example is that users and customers are not satisfied with it. They may find it too difficult to use, unable to support certain user tasks, etc. Another example is that the program doesn't cooperate properly with existing, surrounding software. Unstated user expectations (tacit requirements), misunderstood requirements and misunderstood existing software are typical causes of requirement defects. The requirement defects can relate to functional as well as non-functional requirements. 2. Implementation Defects (the development activities that produce a workable program. Implementation is mainly carried out by programmers. We have an implementation defect if the product doesn't work as intended by the programmers. Typically, implementation defects show up as program crashes or obviously wrong results.
Requirements	Functional	A requirement that specifies a function that a system or system component must be able to perform (IEEE 1999). What the system should do (Sommerville and Sawyer 1997) e.g. it should generate membership numbers for each person joining club etc.
Requirements	Growth/ change	functional and non-functional requirements not documented in original specification that result in changes over time, incorporates changeability decay (Arisholm and Sjoberg 2000)
Requirements	Market Driven	Requirements are sketchy and informal Use of techniques from manufacturers rather than software engineering Specification is in the form of a marketing presentation No readily identifiable 'customer' developers tend to have less experience in application domain. Projects rely on consultants for advice on desirable features Less structured approaches adopted. Task force used in 'brainstorming' sessions (Loucopoulos and Karakostas 1995).
Requirements	Non-functional	Systems quantities or quality attributes. E.g. safety, security, reliability, usability, maintainability, cost and development time (Gross and Yu 2001). High level non-functional requirements often decompose into functional requirements (Sommerville and Sawyer 1997) they are not specifically concerned with the functionality of a system, placing restrictions of the product being developed and the development process.
Requirements	Phase	The period of time in the software life cycle during which the requirements for a software product are defined and documented
Requirements	Poor user understanding	User understanding of their own needs is often confused and undetected until too late – a customer will often ask for functions that are not needed and prove difficult to implement.
Requirements	Process	Processes/activities within the requirements phase of software development that include elicitation, analysis, documentation and validation as well as links to resources, traceability and general requirements management and engineering.

Requirements	Technical Issues (In PPP Project)	Requirements Growth/change; Vague/ambiguous requirements; Requirements Process definition; Poor user understanding; Requirements traceability. (See also 'Technical RE process problems)
Requirements	Qualities	Quality attributes in the requirements process (from (Davis et al. 1993)): Achievable; Annotated by Relative Stability; Annotated by Version; Annoted by Relative Importance; At Right Level of Detail; Complete; Concise; Correct; Cross-Referenced; Design Independent; Electronically Stored; Externally Consistent; Executable/Interpretable; Internally consistent; Modifiable; Not Redundant; Organized; Precise; Reusable; Traceable; Traced; Unambiguous; Understandable; Verifiable
Requirements	Qualities in conceptual modelling	Syntactic Quality; Semantic Quality; Pragmatic Quality (incorporating the concept of feasibility and level of understanding into the modelling process) (Lindland et al. 1994).
Requirements	Review	A process or meeting during which the requirements for a system hardware item, or software item are presented to project personnel, managers, users, customers, or other interested parties for comment or approval. Types include system requirements review, software requirements review (IEEE 1999).
Requirements	Specification	A document that specifies the requirements for a system or component. Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards (IEEE 1999).
Requirements	Tacit	Unstated requirements (see requirements defects).
Requirements	Traceability	A link or definable relationship between entities (Watkins and Neal 1994) that relates primarily to the requirements stage of software development.
Requirements	Vague/ ambiguous	Requirement documentation is incomplete and flawed. Also called requirements uncertainty (Moynihan 2000).
Requirements Engineering	(See also Engineering – Requirements)	A separate field of investigation and practice established in mid 1970s (Loucopoulos and Karakostas 1995). The science and discipline concerned with analysing and documenting requirements, including needs analysis, requirements analysis, and requirements specification. It also provides the appropriate mechanisms to facilitate the analysis, documentation, and verification activities. Requirements engineering can also be defined as a combination of requirements analysis and the documentation of the requirements into a form called requirements specifications. Chapter 1 (Thayer and Dorfman 1990).
Requirements Engineering	Organisational Issues (In PPP Project)	Practitioner communication; Resources; Skills; Staff retention; Training; User communication. (See also Organisational RE Process problems)
Requirements Engineering Process	RE Process	Activities which attempt to understand the exact needs of the users of the software intensive system and to translate such needs into precise and unambiguous statements which will subsequently be used in the development of the system. (Loucopoulos and Karakostas). Activities performed in the requirements phase that culminate in producing a document containing the software requirements specification (Jalote 1997). The set of activities required to gather, specify, validate and engineer a set of requirements (Britton 2000) and (IEEE Software - Thayer and Dorfman 1990, page 1).
Resources	In PPP project	This relates to time, costs, investment in tools and people. Timescales and estimates given at beginning of project to be managed with allocation of adequate resources (staff time/training/costs of new tools) to include long-term software improvement activities.
Semantics	(as used in modelling)	The branch of linguistics and logic concerned with meaning (Concise Oxford Dictionary 1999)
Skills	In PPP project	Level of spread and appropriate expertise available to prevent over-dependence on few experienced staff. Sharing of best practice
Software	Engineering	technical, managerial activities carried out in the production of software (Madhavji 1991). To include: determination and specification of system and software requirements; analysis and management of risk; software prototyping; design, implementation; verification and validation; software quality control and assurance; integration of components; documentation; management of software configurations and versions, management of data, evolution of software; project management; software evaluation; software contracting; software acquisition; commissioning and decommissioning of software. Or: An engineering discipline that applies sound scientific, mathematical, management, and engineering principles to the successful building of large computer programs (software) (Dorfman and Thayer 1997) Or as SEI CMMI define: Software engineering covers the development of software systems. Software engineers focus on applying systematic, disciplined, and quantifiable approaches to the development, operation, and maintenance of software. When you select software engineering for your model, the model will contain the Process Management, Project Management, Support, and Engineering process areas. Discipline amplifications specific to software engineering are provided to help you interpret specific practices for software engineering (SEI 2002).
Software Requirements Review	(SRR)	A review of the requirements specified for one or more software configuration items to evaluate their responsiveness to and interpretation of the system requirements and to determine whether they form a satisfactory basis for proceeding into preliminary design of the configuration items (IEEE 1999).

Software requirements specification	(SRS)	Documentation of the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces (IEEE Std 1012-1986 [12]) (IEEE 1999). A document that describes all the externally observable behaviours and characteristics expected of a software system (Davis et al. 1993).
Specification	Requirements	A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behaviour, or other characteristics of a system or component, and often, the procedures for determining whether these provisions have been satisfied (IEEE 1999).
Staff Retention	In PPP project	Incorporates recruitment and workforce stability. Recruiting staff of the right level and retaining experienced staff.
Stakeholders		All practitioners and customers, and users – all people affected by the system with direct or indirect influence on the system requirements (Sommerville and Sawyer 1997)
Supplier		The person, or persons, who produce a product for a customer. In the context of this study and the IEEE (1998) recommended practice, the customer and the supplier may be members of the same organization.
Supplier of System		System developer or service provide who delivers a solution to meet the expected level of functionality and ensure successful integration of the technical system in the organizational setting (Loucopoulos and Karakostas 1995)
Syntax		A set of rules ...the structure of statements in a computer language (Concise Oxford Dictionary 1999)
System		A collection of components organized to accomplish a specific function or set of functions (IEEE 1999).
Systems	Engineering	an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem...(International Council on Engineering Systems (INCOSE 1999) in (Leffingwell and Widrig 2000) p.58 or as SEI define as: Systems engineering covers the development of total systems, whichmay or may not include software. Systems engineers focus on transforming customer needs, expectations, and constraints into product solutions and supporting these product solutions throughout the life of the product (SEI 2002).
System Requirements Engineering		Is the science and discipline concerned with analysing and documenting system requirements. It involves transforming an operational need into a system description, system performance parameters, and a system configuration through the use of an iterative process of analysis, design, trade-off studies and prototyping Chapter 1: (Thayer and Dorfman 1990).
Technical RE process problems	In PPP study	A class of problem that directly influences the production of the RE Specification, more usually the responsibility of Developers and Engineers, to include: Complexity of application; Requirements growth/change; vague requirements; requirements process definition; poor user understanding; requirements traceability.
Traceability		See 'Requirements' traceability
Training		Training needs both in technical and organisational areas
User		The individual, group or organization that will work with the system itself (Loucopoulos and Karakostas 1995).
User Communication		Supplier communication with users (e.g. how company structure dictates who discusses customer requirement needs with the customer and user).

Appendix B: Profiles of Companies involved in the PPP Focus Groups

Company number	HW/SW Producer	UK or Multi-national?	Size (people)	SE size (people)	Age (yrs)	SW type	CMM Level
1	HW/SW	MN	>2000	>2000	>50	RT/EM	1*
2	SW	UK	100-500	100-500	20-50	Bus	1
3	HW/SW	MN	>2000	500-2000	>50	RT/EM	1
4	HW/SW	MN	>2000	500-2000	>50	RT/EM	1
5	SW	MN	>2000	>2000	10-20	RT	4*
6	SW	MN	>2000	>2000	10-20	RT	3*
7	SW	MN	>2000	>2000	20-50	Packs	1
8	SW	UK	10-100	10-100	5-10	Bus	2
9	SW	MN	10-100	10-100	10-20	RT/EM	3
10	SW	MN	>2000	10-100	10-20	Sys/EM	1
11	HW/SW	MN	500-2000	11-25	20-50	RT/EM	2
12	A pilot study not used in this report – each focus group comprised a mixture of practitioner roles and did not conform to the role-specific structure of the other 12 companies						
13	SW	UK	100	40	10-20	Bus	3

Key: RT = Real Time; EM = Embedded; Bus = Business systems; Packs = Packages; Sys = Systems software

* Formally CMM assessed. Companies without * are all undertaking SPI and have self-estimated their CMM levels through answering questions inmy questionnaire. (See Appendix G for further details of the self-assessment.)

Appendix C: Audit of self-assessed SW CMM Level Companies

The 9 companies in the PPP study who self-assessed their maturity levels comprise:

- 5 Level 1 SW-CMM
- 2 Level 2 SW- CMM
- 2 Level 3 SW-CMM

1. SW-CMM Level 1 self-assessed companies. These companies have the lowest process maturity: interviews with practitioners and examination of the detailed questionnaire reveal ad-hoc processes with little standardisation and a lack of formal documentation.
2. SW-CMM Level 2 self-assessed companies. These companies have external quality audits to include ISO certification (9001, 9002), SPICE and TickIT. They have a formal approach to documenting and defining software development processes. The SPI programme has been in operation for more than 5 years. Senior management is committing resources to the SPI effort, e.g. training, staff and tools. The practices employed by these companies such as project post mortems, inspections and reviews represent repeatable key process areas (KPAs) within the Level 2 SW-CMM.
3. SW-CMM Level 3 self-assessed companies. These companies have external quality audits ISO 9001 and TickIT that span many years. They have all the key processes of the level 2 companies with the additional features: They are preparing for formal SW-CMM assessment, have processes in place for cross-project communication. Company-wide process standards are in place and are used. Formal data is collected on improvement effort; practitioners are fully involved from the beginning of the process improvement programme. Groups have been set up specifically to support the software improvement effort. SPI is driven by management and dedicated staff and is very well resourced.

All self-assessed companies have undergone an independent assessment by the PPP team. This involved:

a) visiting all companies
b) examining documentation
c) re-examining questionnaire
d) discussing SPI with quality managers and individual focus groups
e) comparing the companies' formal quality certification (e.g. ISO) with SW-CMM level KPAs

I have used quality standards such as the ISO 9000 series as one measure of SW-CMM process maturity. Although the ISO does not link directly to higher levels of the CMM, they both share a common concern with quality and process management. There is a strong correlation between ISO 9001 and the SW-CMM, e.g. defining and documenting: responsibility and authority, internal quality audits, interrelation, company-wide training 'of all personnel performing activities affecting quality', peer reviews, defining organizational and technical interfaces between different groups' (Tingey 1997). The biggest difference between the two quality assessments is the emphasis of SW-CMM on continuous process improvement (Paulk 1994). However, if a company retains ISO certification over many years and has dedicated SPI staff it is likely that they are not taking a snap-shot view of process improvement.

An organisation that is ISO 9001 compliant has significant process strengths at level 2 and noticeable strengths at level 3. There is also a suggestion that if a company retains ISO certification over a period of time its processes become more predictable as they mature. If an organization is following "the spirit of ISO 9001" it seems probable the organization would be near or above level 2 (Paulk 1994).

While it is difficult to prove that a company is following 'the spirit' of ISO 9001,my mix of quantitative and qualitative data gathering supports the levels attributed to the companies. Company profiles are given to show formal certification and practices of the 4 companies who have self-assessed their process maturity to be above the ad-hoc CMM level 1:

Level 2 self-estimated company profiles:

Company 8: Questionnaire completed by Implementation Services Manager, who is a member of the SEI and ASQ (assisted by member of research team).

Indicators consistent with Level 2 process maturity:

- Company documents and defines software processes
- Improvement programme has been in operation for more than 5 years
- The company refers to the CMM and SPICE and use the materials to guide them in their SPI programme
- The goals of the process improvement programme are congruent with the goals of the company
- process improvement programme is very well tailored to the needs of the company

CMM Level Self-ass	ISO 9001	ISO 9002	SPICE	TickIT	Over 5 years	Practices
2	✓	✓				reviews; standards & procedures; training & mentoring; project post mortems; metrics/data collection; estimating tools; automated tools
Our confidence level in CMM assessment: HIGH						

Company 11: Questionnaire completed by the quality manager who is a member of British Computer Society (assisted by member of research team).

Indicators consistent with Level 2 process maturity:

- The company does not have a formal company-wide programme, suggesting that the company does not have a level 3 'defined' process maturity
- They have a formal approach to documenting and defining software development processes
- The Process Improvement programme has been in operation for more than 5 years
- The goals of the process improvement programme are not totally congruent with overall company goals yet they accept that congruence is very important
- The company is aware of the need to tailor the process improvement programme to the needs of the company but that this is only 'fairly well' tailored currently
- Senior management is very committed to the process improvement programme

CMM Level Self-ass	ISO 9001	ISO 9002	SPICE	TickIT	Over 5 years	Practices
2	✓	✓	✓	*(1-3 yrs)*	✓	Inspections; Standards and procedures; Testing; Project Management; Project Post Mortems; Reviews; Metrics/data collection; Risk assessment. Training/ Mentoring; getting buy-in at the beginning of building procedures; Internal Leadership; Internal process ownership (varies in each dept);
Our confidence level in CMM assessment: HIGH						

Level 3 self-estimated CMM Level Companies

Company 9: Questionnaire filled in by quality manager (assisted by member of research team)

Indicators consistent with level 3 process capability:

- SPI programme in place for over 5 years
- Formal approach to documenting processes in software development in place
- Objectives and goals of the process improvement programme are clearly stated and are fully congruent with company goals
- planning to use CMM formally
- Extensive research on different approaches to process improvement undertaken
- SPI is very well tailored to the needs of the company
- Senior management is totally committed to SPI
- Design Authorities as part of the SPI initiative set up to improve communication between teams. (This cross-project communication is a key feature of the CMM 'defined' organisational level 3)

Company 9 demonstrates both documenting and using practices associated with a level 3 process maturity company.

ISO 9001	TickIT	Over 5 years	Practices	SPI Data Collection	SPI Resources
✓	✓	✓	Audits and Reviews; Standards and procedures; project post mortems (brought about many changes); risk assessment; estimating tools; automated tools; metrics/data collection; Inspections. Training/Mentoring; Internal leadership; Internal process ownership; feedback from engineeers (forum)	requirements management; software project planning; software tracking and oversight; software subcontract management; software quality assurance; software configuration management; training programme; peer reviews; fault analysis.	Executive support; experienced staff; Driven by highly respected staff with one person dedicated to SPI. Clear responsibilities assigned to SPI team. SPI Groups: Software Engineering Process Group; Software Process Action Team; Quality Team; Software Configuration Management Group; Documentation Support Group.
Our Confidence in Self-Assessment: MEDIUM level of confidence that company is Level 3					

Company 13: Questionnaire completed by quality manager (assisted by member of research team).

Indicators consistent with level 3 process capability:

- Has a formal documented process approach to software development
- Process improvement programme objectives and goals are clearly stated and are fairly congruent with company's goals

- Design Authorities as part of the SPI initiative serve to improve communication between teams

Company 13 demonstrates a greater process maturity that the level 2 companies as many of the KPAs of level 3 and some of level 4 have been implemented – showing a defined and partially managed level of maturity.

ISO 9001	TickIT	Over 5 years	Practices	SPI Data Collection	SPI Resources
✓	✓	✓	Reviews; standards and procedures; project post mortems; metrics/data collection; risk assessment; estimating tools; automated tools; Inspections Training/Mentoring (lots at start); Reward schemes; promoting internal leadership; Internal process ownership; Stringent Control	requirements management; software project planning; software tracking and oversight; software subcontract management; software quality assurance; software configuration management; training programme; peer reviews; fault analysis.	Executive Support; Experienced staff. SPI groups: Quality council for business as a whole, sub-group of people for SPI; Software configuration management group; documentation support group. More than 2 (part-time) staff are dedicated to process improvement. The process improvement programme team is independent of the software developers with clear responsibilities assigned to SPI.
Our Confidence in Self-Assessment: MEDIUM level of confidence that company is Level 3					

Appendix D: The Expert Panel

Name of Participant	Current/most recent company	Position/relevant experience
Bangert, A		IT Consultant
Childs, P	B C Electrical Techniques Ltd	IT Consultant
Fox, D	Clerical Medical Investment Group	IS Project Manager
Homan, D	NORTEL (ex)	Quality Manager
Hough, A	Moneyfacts Group Plc	IT Director
Kujala, S (PhD)	Helsinki University of Technology	Senior Researcher (involved in assessment of the REAIMS model)
Kutar, M (PhD)	University of Hertfordshire	Lecturer (expert in Requirements methods)
Maiden, N (PhD)	City University, London	Head of Research Centre
McBride, T	University of Technology, Sydney	Lecturer/trained SPICE assessor, trained ISO 9001 auditor/ ex chairman of NSW Software Quality Assoc, on ISO sub-committee to develop software engineering standards
Nuseibeh, B (PhD)	Open University, Computing Dept	Professor (Requirements)
Anonymous	Insurance Company	IT Business Analyst - Requirements
Robinson, J (PhD)	Rand, USA	Senior Information Scientist (many years experience as software requirements Engineer)
Sawyer, P (PhD)	Lancaster University	Head of Computing Dept, co-author of Text Book on Requirements Engineering
Smith, R	CSE International Ltd	Consultant
Spooner, A	Norwich Union	Web Development Manager/Project manager
Steele, J	BAe Systems	Head of Hardware Engineering
Stephens, M		Senior Information Analyst
Sutcliffe, A (PhD)	UMIST Dept of Computation	Professor (Requirements)
Wilkinson, V	SEMA (ex)	Analyst/Programmer
Wilson, D (PhD)	University of Technology, Sydney	Associate Professor

Appendix E: Expert Panel Questionnaire (refers to documentation in Appendix F)

University of Hertfordshire

Centre for Empirical Software Process Research

Expert Panel Questionnaire

Validating a Requirements Process Improvement Model

Please return completed questionnaire to:

Sarah Beecham, Department of Computer Science, University of Hertfordshire
College Lane, Hatfield, Hertfordshire A10 9AB.

If, for any reason, you feel unable to complete this questionnaire then please also return it to the above address.

Demographics

This section is concerned with information about you and your related software development interests. This information will be treated in the STRICTEST CONDFIDENCE and any publication of this study will present information in aggregate form and such information will be anonymous and unattributable to individual organisations or individual respondents. Your personal details will not be passed on to any third party.

(Please print your personal details below)

Name: --

Company: ---

Address: ---

E-mail: -- Job Title: --

Q1 How long have you been involved in software development?

a) 5 years or less [ ]   b) 6 – 10 years [ ]   c) 11 – 20 years [ ]   d) Over 20 years [ ]

Q2 The requirements software process

	none (1)	fair (2)	good (3)	expert (4)	no opinion
a) How do you rate your knowledge of the requirements engineering process?	[ ]	[ ]	[ ]	[ ]	[ ]
b) How do you rate your academic involvement in requirements engineering?	[ ]	[ ]	[ ]	[ ]	[ ]
c) How do you rate your practical experience in participating in the requirements phase of software engineering?	[ ]	[ ]	[ ]	[ ]	[ ]

Q3 Please indicate the strength of your agreement/disagreement with the following statements:

	strongly disagree (1)	disagree (2)	agree (3)	strongly agree (4)	no opinion
a) There is evidence to suggest that companies have problems with their requirements process	[ ]	[ ]	[ ]	[ ]	[ ]
b) The requirements process is giving companies more problems than other parts of software development	[ ]	[ ]	[ ]	[ ]	[ ]
c) In general, the requirements process is in need of improvement	[ ]	[ ]	[ ]	[ ]	[ ]

Q4 The Software Capability Maturity Model (CMM)

	none (1)	fair (2)	good (3)	expert (4)	no opinion
a) How do you rate your knowledge of the CMM?	[ ]	[ ]	[ ]	[ ]	[ ]
b) How do you rate your academic involvement in the CMM – in terms of active research?	[ ]	[ ]	[ ]	[ ]	[ ]
c) How do you rate your practical experience of the CMM?	[ ]	[ ]	[ ]	[ ]	[ ]

d) Have you or your company been involved in a CMM assessment? (Please tick/fill in one of the five options)

1. No [ ]
2. Assessment in progress [ ]
3. Informal CMM assessment made [ ] 3.1 CMM level attained [ ]
4. Formal CMM assessment made [ ] 4.1 CMM level attained [ ]
5. Other [ ]
   Please Note:

Q5 Please indicate the strength of agreement/disagreement with the following statements:

	strongly disagree (1)	disagree (2)	agree (3)	strongly agree (4)	no opinion
a) The CMM clearly defines software process activities	[ ]	[ ]	[ ]	[ ]	[ ]
b) The CMM 5 maturity levels helps companies to prioritise process implementation	[ ]	[ ]	[ ]	[ ]	[ ]
c) The CMM framework can be tailored to suit a company's specific needs	[ ]	[ ]	[ ]	[ ]	[ ]
d) The guidelines in the CMM represent current best practice in software development	[ ]	[ ]	[ ]	[ ]	[ ]

Please refer to the 'accompanying documentation' booklet to answer the following sections

Model Framework Design

This section deals with the structure and context of the Requirements CMM framework.

Q6 Please refer to <u>Document 1</u> before answering the questions below.

The questions in this section should be answered using a scale of 1 to 4, where 1 is "not at all", and 4 is "very". 2 relates to "fairly or limited" and 3 is "acceptable" or "good but could be improved".

If you have no opinion or the question is not relevant to you, please put a tick in the last column.

	Not at all (1)	(2)	(3)	very (4)	no opinion
a) How appropriate is it to adapt the Software CMM Maturity level characteristics to create maturity goals for the requirements CMM?	[ ]	[ ]	[ ]	[ ]	[ ]
b) How appropriate is the level of detail in the Requirements CMM for an initial guide to the requirements process?	[ ]	[ ]	[ ]	[ ]	[ ]
c) How complete is the Requirements-CMM framework?	[ ]	[ ]	[ ]	[ ]	[ ]
d) How clear is the presentation of the Requirements-CMM?	[ ]	[ ]	[ ]	[ ]	[ ]
e) How consistent is the level of detail given within the Requirements CMM?	[ ]	[ ]	[ ]	[ ]	[ ]

	no knowledge (1)	fair knowledge (2)	good knowledge (3)	expert knowledge (4)	no opinion
f) How much previous knowledge of the SEI's CMM do you think you need to be able to interpret this high level framework?	[ ]	[ ]	[ ]	[ ]	[ ]

Developing a Level 2 Requirements Capability Maturity Model

This section deals with the requirements CMM at Level 2 capability.

Q7. Level 2 CMM 'Design and structure'

Please look at Document 2 and rate your response to the following questions:

	Not at all (1)	(2)	(3)	very (4)	no opinion
a) How clear is this presentation?	[ ]	[ ]	[ ]	[ ]	[ ]
b) How easy is it to understand the path from initial goal to question to final process?	[ ]	[ ]	[ ]	[ ]	[ ]
c) How easy would it be to adapt this framework? (e.g. {add/remove/amend} goals, questions and processes?)	[ ]	[ ]	[ ]	[ ]	[ ]
d) How well does the level 2 model in Document 2 relate to the higher level model in Document 1? (Please refer back to Document 1 for this)	[ ]	[ ]	[ ]	[ ]	[ ]

	Too few ← (1)	(2)	Correct number (3)	(4)	→ Too many (5)	no opinion
e) We have divided the requirements phase of software engineering into 20 key processes (P1-P20 in Dcoument 2). Is this a good number or would the model benefit from a more comprehensive list – with more processes? Or would having fewer processes enhance the design?)	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Q8 Level 2 CMM 'Content and meaning'

Please continue to look at Document 2 to rate your assessment of the following questions:

	not at all (1)	(2)	(3)	very (4)	no opinion
a) How appropriate is the level of detail for an introduction to requirements processes? (Level 2 'introduces' requirements baseline processes)	[ ]	[ ]	[ ]	[ ]	[ ]
b) How appropriate is it to include organisational processes (e.g. skills audit) and technical processes (e.g. techniques to trace requirements) in one model?	[ ]	[ ]	[ ]	[ ]	[ ]
c) How appropriate is it to model these 20 processes in one framework?	[ ]	[ ]	[ ]	[ ]	[ ]
d) How appropriate is it to decompose the requirements process into these 5 questions (Q1 - Q5)?	[ ]	[ ]	[ ]	[ ]	[ ]
e) How well do the questions (Q1 - Q5) relate to the Level 2 goal?	[ ]	[ ]	[ ]	[ ]	[ ]
f) How well do questions (Q1 – Q5) help towards focussing on individual requirements processes (P1 - P20)?	[ ]	[ ]	[ ]	[ ]	[ ]
g) How well do questions (Q1 - Q5) cover all the key activities involved in the requirement stage of software development?	[ ]	[ ]	[ ]	[ ]	[ ]

Level 2 CMM: 'Usefulness of Requirements Processes'

Below we list the processes (P1 – P20) that represent our 'candidate' key processes for a Level 2 compliant organization (as modeled in Document 2). Please note that Level 2 maturity reflects best practice at a 'Project level' and not at an organisation-wide level.

Q9 Please categorize the processes listed below as follows:

Not-needed (N) | Desirable (D) | Essential (E) | Don't know (?)

Process Description reflecting a Level 2 Requirements Capability	Priority N/D/E/?
P1: Follow a written organisational policy for managing the system requirements allocated to the software project	
P2: Establish project responsibility for analysing the system requirements and allocating them to hardware, software, and other system components	
P3: Implement training programme to recognise and meet technical and organisational requirements needs within the project	
P4: Establish process to identify stakeholders within the requirements phase of the project	
P5: Provide adequate resources and funding for managing the allocated requirements in the project	
P6: Establish process to identify skills needs within project	
P7: Institute process to maintain organisational stability within project, e.g. control staff change	
P8: Explore alternative solutions, requirements techniques and tools for the project	
P9: Establish/maintain process to reach agreement with customer on requirements for project	
P10: Establish/maintain process to involve key stakeholders in requirements phase of project	
P11: Set realistic improvement goals to address problems in the requirements process project	
P12: Establish/implement process to assess feasibility & external environment of project	
P13: Establish/maintain repeatable requirement traceability process that is project-based	
P14: Establish a repeatable process to manage complex requirements at project level	
P15: Establish a repeatable process to manage vague requirements at project level	
P16: Establish a repeatable process to manage requirements growth at project level	
P17: Establish a repeatable process to manage user understanding at project level	
P18: Monitor progress of the set requirements goals	
P19: Agree and document technical and organisational attributes specific to project	
P20: Establish a process to review allocated requirements within the project to include software managers and other affected groups	

Q10 Please indicate the strength of agreement/disagreement with the following statements:

	strongly disagree (1)	disagree (2)	agree (3)	strongly agree (4)	no opinion
a) All processes (P1-P20) work together to achieve requirements process improvement at a repeatable level	[ ]	[ ]	[ ]	[ ]	[ ]
b) processes can be implemented gradually (one-by-one)	[ ]	[ ]	[ ]	[ ]	[ ]
c) The requirements process can be improved by implementing the individual processes in any order.	[ ]	[ ]	[ ]	[ ]	[ ]
d) Each individual process is easy to understand, i.e.P1 – P20 are clearly defined and unambiguous	[ ]	[ ]	[ ]	[ ]	[ ]
e) All key processes are represented	[ ]	[ ]	[ ]	[ ]	[ ]
f) Each process relates to requirements engineering activities	[ ]	[ ]	[ ]	[ ]	[ ]
g) Each process relates to Maturity Level 2	[ ]	[ ]	[ ]	[ ]	[ ]
h) Processes can be incorporated into a software process improvement programme	[ ]	[ ]	[ ]	[ ]	[ ]
i) It would be possible to extend each process to create specific guidelines and prescriptions, i.e. convert process guidelines into practice.	[ ]	[ ]	[ ]	[ ]	[ ]
j) All processes listed are at a similar level of abstraction (e.g. no process could be considered a 'part' of another process in the same group)	[ ]	[ ]	[ ]	[ ]	[ ]

Level 2 CMM 'Ease of understanding':

Q11 Please refer to <u>Documents 2 and 3</u> and indicate the strength of agreement/ disagreement with the following statements:

	strongly disagree (1)	disagree (2)	agree (3)	strongly agree (4)	no opinion
a) Separating requirements into 5 phases (Q1 – Q5) as presented in Document 3 helps to relate requirements processes to general software development practices	[ ]	[ ]	[ ]	[ ]	[ ]
b) Viewing requirements in 5 stages helps practitioners to understand when to implement each process	[ ]	[ ]	[ ]	[ ]	[ ]
c) This view of requirements helps to incorporate the processes into a company's general improvement programme	[ ]	[ ]	[ ]	[ ]	[ ]
d) Viewing requirements in these 5 phases is a reflection of how requirements are implemented in practice	[ ]	[ ]	[ ]	[ ]	[ ]
e) The set of processes listed in each of the 5 phases reflects the activities associated within each phase, e.g. the 'Identify Stakeholders' process (P4) is an activity that belongs to the requirements 'elicitation' phase (Q2)	[ ]	[ ]	[ ]	[ ]	[ ]

	Not enough detail (1)	(2)	The right amount of detail (3)	(4)	Too much detail (5)	no opinion
f) There is sufficient detail in Document 3 to guide the user towards recognising baseline requirements processes	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Level 2 CMM 'Process Guidelines'

Document 4 gives an example of a guideline for the "Identify Stakeholders" requirement process (Process P4). Guidelines are an extension of the processes listed in Document 2 (processes P1-P20).

Q12 Please refer to <u>Document 4</u> and indicate the strength of agreement/disagreement with the following statements (some statements ask you look at Document 2 as well):

	strongly disagree (1)	disagree (2)	agree (3)	strongly agree (4)	no opinion
a) It is easy to understand how the sub-goal in Document 4 relates to Process (P4) in the Document 2.	[ ]	[ ]	[ ]	[ ]	[ ]
b) The guidelines given are complete at this level of detail	[ ]	[ ]	[ ]	[ ]	[ ]
c) The guidelines given are relevant to requirements engineering activities	[ ]	[ ]	[ ]	[ ]	[ ]
d) The guidelines are at the same level of granularity	[ ]	[ ]	[ ]	[ ]	[ ]
e) Dividing this process into smaller activities in this way will help practitioners to implement the process	[ ]	[ ]	[ ]	[ ]	[ ]
f) Decomposing this process into smaller activities will help companies to analyse where the process needs strengthening	[ ]	[ ]	[ ]	[ ]	[ ]
g) The activities given in the guidelines are clear and easy to understand	[ ]	[ ]	[ ]	[ ]	[ ]
h) The activities given are general and likely to apply to most companies	[ ]	[ ]	[ ]	[ ]	[ ]
i) It would be helpful to provide guidelines for all processes listed in the Level 2 model (Document 2)	[ ]	[ ]	[ ]	[ ]	[ ]

Level 2 CMM Assessment

We appreciate that your answers in this section will be based on your 'impression/perception' as you have not had the opportunity to use the assessment model in practice.

Q13 Please look at <u>Document 5</u> before answering the following questions:

	not at all (1)	(2)	(3)	very (4)	no opinion
a) How appropriate is it to use the scoring matrix (Table 1) in Document 5 to reflect general requirements process capability?	[ ]	[ ]	[ ]	[ ]	[ ]
b) How appropriate is it to assess requirements by examining the 3 dimensions "Approach, Deployment, and Results"?	[ ]	[ ]	[ ]	[ ]	[ ]
c) How appropriate is this method for a Level 2 capability assessment (i.e. assessing whether the requirements process is repeatable)?	[ ]	[ ]	[ ]	[ ]	[ ]
d) How easy is it to use this assessment method?	[ ]	[ ]	[ ]	[ ]	[ ]
e) How suited is the assessment method for all requirement stakeholders to use?	[ ]	[ ]	[ ]	[ ]	[ ]
f) How easy would it be to adapt this assessment method to meet individual company needs? (i.e. a company may identify other processes in need of measurement and may also require different measurement criteria)	[ ]	[ ]	[ ]	[ ]	[ ]

Q14 Please indicate the strength of agreement/disagreement with the following statements relating to <u>Document 5</u>

	strongly disagree (1)	disagree (2)	agree (3)	strongly agree (4)	no opinion
a) This assessment method could be applied to all the processes presented in the Level 2 Requirements model (Document 3)	[ ]	[ ]	[ ]	[ ]	[ ]
b) It is appropriate to relate this assessment method to all 5 CMM Levels	[ ]	[ ]	[ ]	[ ]	[ ]
c) The 3 dimensions (approach, deployment, results) in the scoring matrix cover the key areas to measure	[ ]	[ ]	[ ]	[ ]	[ ]
d) The evaluation clauses in Table 1 within the 3 dimensions are clear/easy to interpret	[ ]	[ ]	[ ]	[ ]	[ ]
e) The scoring matrix is at the right level of detail to give effective results	[ ]	[ ]	[ ]	[ ]	[ ]
f) This assessment method retains the CMM level concept	[ ]	[ ]	[ ]	[ ]	[ ]
g) The assessment method is self-explanatory and requires no further examples/definitions to be used effectively	[ ]	[ ]	[ ]	[ ]	[ ]
h) The method used is general and likely to apply to most companies	[ ]	[ ]	[ ]	[ ]	[ ]
i) This method of highlighting process weaknesses will assist managers to prioritise their improvement activities	[ ]	[ ]	[ ]	[ ]	[ ]

Overall Impression of the Requirements CMM

Q15 Having looked at all the documentation, we now ask you for your overall impression of the model:

	not at all (1)	(2)	(3)	very (4)	no opinion
a) How well does the new requirements framework retain the CMM concept?	[ ]	[ ]	[ ]	[ ]	[ ]
b) How complete is this initial requirements model framework?	[ ]	[ ]	[ ]	[ ]	[ ]
c) How useful is it to take a process view of requirements to improve the overall requirements process?	[ ]	[ ]	[ ]	[ ]	[ ]
d) How appropriate is it to aim requirements improvement activities towards satisfying maturity level goals?	[ ]	[ ]	[ ]	[ ]	[ ]
e) How clear is the overall presentation of the model?	[ ]	[ ]	[ ]	[ ]	[ ]
f) How adaptable is this model? (i.e. ease with which the model can be adapted to guide users towards further key processes required to achieve higher level maturity goals)	[ ]	[ ]	[ ]	[ ]	[ ]
g) How useful would it be to the software industry to continue to develop this model?	[ ]	[ ]	[ ]	[ ]	[ ]
h) How useful would it be to the research community to continue to develop this model?	[ ]	[ ]	[ ]	[ ]	[ ]
i) How realistic is it to ask companies to look at their requirements process in this piecemeal way?	[ ]	[ ]	[ ]	[ ]	[ ]

	Not enough (1)	(2)	Correct level (3)	(4)	Too much (5)	No opinion
j) Has the level of detail provided by this questionnaire allowed you to give a fair assessment of the strengths and weaknesses of the new Requirements CMM?	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

May we contact you by e-mail if we need to follow up any of your responses to this questionnaire? Yes / No

Please use the next page to make any further comments you may have.

Further Comments:

Thank you for helping to validate this Requirements Capability Maturity Model.

Thank you for completing the questionnaire

Please return the questionnaire to:

Sarah Beecham
Department of Computer Science
University of Hertfordshire
College Lane
Hatfield, Herts
AL10 9AB

Appendix F: Documentation Booklet sent to Experts to accompany the Questionnaire (given in Appendix E)

University of Hertfordshire

Centre for Empirical Software Process Research

Accompanying Documentation

Introducing a Requirements Process Improvement Model

This documentation is designed for use with the Expert Panel Questionnaire "Validating a Requirements Process Improvement Model"

If you require any further information please e-mail s.beecham@herts.ac.uk.

Contents

Model Overview

The Requirements Capability Maturity Model (Requirements CMM)

We have adapted the well established Software CMM improvement framework to represent only the requirements process. This new 'Requirements CMM' is designed to help practitioners strengthen the requirements process within software development. The model uses the CMM framework to identify key requirements processes and prioritise their implementation. The CMM is the subject of much on-going research and is also well-resourced in terms of support, continual improvement and on-going assessment. It is our intention that the Requirements CMM is used in conjunction with an ongoing software improvement programme, particularly an improvement programme that uses the Software CMM framework.

Requirements CMM in context

The Requirements CMM is intended to be used by practitioners familiar with the Software CMM maturity concept. However, it should be possible to use the model independently of the Software CMM to assess requirements process capability.

A Goal/Question/Metric approach

Identifying goals prior to following guidelines helps process improvement. We formalise this goal-focussed view of requirements by using a Goal/Question/Metric (GQM) approach. How the GQM approach helps to control the setting and subsequent fulfilment of goals is demonstrated in Documents 2 and 4.

Requirements CMM Goals:
We use the Software CMM framework to provide us with maturity goals for the requirements process (Document 1). Each goal relates to a level of capability, for example the Level 2 goal is: "Implement a repeatable requirements process" (Document 1 and 2).

Requirements CMM Questions:
Assessing whether a Level 2 requirements goal has been achieved requires addressing 5 questions. These 5 questions relate directly to requirements phases: management; elicitation; analysis and negotiation; documentation; and validation (Documents 2 and 3). These 5 requirements phases are not to be confused with the 5 process maturity stages modelled in the CMM framework (i.e. ad hoc; repeatable; defined; managed; and optimised) in Document 1.

Requirements CMM Processes:
Each requirements phase is associated with a set of processes as listed in Documents 2 and 3. We have identified 20 requirements processes that we believe are key to establishing a Level 2 capability. These key processes (P1 – P20) are at a fairly high level of detail, so we plan to assist implementation of each process by providing a process guideline for each process. An example of a guideline is given in Document 4.

Requirements CMM Metrics:
Each of the 20 processes is measured in turn to assess how well the process has been implemented in practice. An example of this form of process assessment is given in Document 5. We have adapted a procedure that has been tried and tested in a high maturity company. This form of assessment can be extended to assess the strength of sub-processes as listed in the guidelines in Document 4 should a finer grained analysis be required.

Validation scope

This validation focuses on the Level 2 model. Maturity Levels 1, 3, 4 and 5 are covered in concept only (Document 1). This is because Level 1 characterises 'ad hoc' practices and has no associated goals and key processes. We are therefore not interested in developing the Level 1 model any further. The higher Levels 3, 4 and 5 are dependent on and build on Level 2 processes. Therefore, we need to know how this baseline model is received before developing higher level maturity models and key processes any further. If however you are interested in viewing draft requirements models at Level 3, 4 and 5 maturity, please refer to our web-page: http://homepages.feis.herts.ac.uk/~pppgroup/requirements_cmm.htm.

Sources

The Level 2 processes presented in the Requirements CMM are from three sources:

1. Our empirical research (Hall *et al.*, 2002; Beecham *et al.*, 2003);
2. The Software CMM (associated key requirements best practices);
3. The requirements literature (supporting our findings and CMM practices).

The model does not cover every requirements process, only those highlighted by practitioners in our empirical study and modelled in the software CMM as being 'key' to the software process improvement.

References

Beecham, S., Hall, T. and Rainer, A. 2003. Software Process Improvement Problems in 12 Software Companies: An Empirical Analysis. Empirical Software Engineering, 8 (1): 7-42.

Hall, T., Beecham, S. and Rainer, A. 2002. Requirements Problems in Twelve Companies: An Empirical Analysis. IEE Proceedings for Software, October, 149: No.5: 153-160.

Paulk, M. C., Weber, C. V., Curtis, B. and Chrissis, M. B. 1995. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, Massachusetts: Addison Wesley Longman, Inc.

Further information

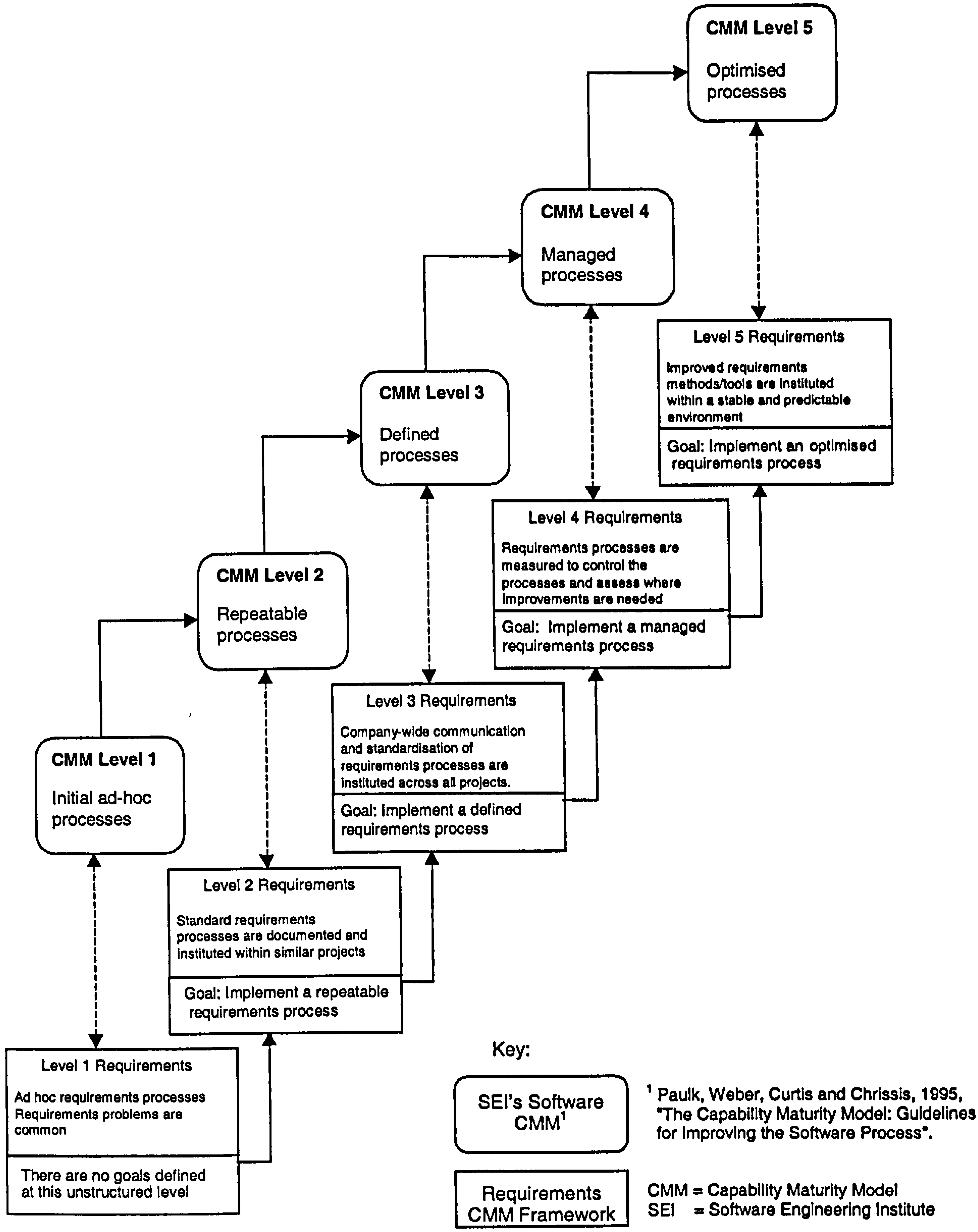
If you want further details relating to sources, references and definitions of terms please refer to:

http://homepages.feis.herts.ac.uk/~pppgroup/requirements_cmm.htm.

If you need any other information please contact Sarah Beecham by e-mail:

s.beecham@herts.ac.uk.

Document 1: Model Framework



Document 2: Level 2 goal-focussed requirements processes

GOAL

Level 2 Requirements Goal

Implement a repeatable requirement process

QUESTION

Q1 How repeatable is your requirements management process?
Processes: P1, P2, P3, P4, P5, P7

Q2 How repeatable is your elicitation process?
Processes: P6, P8, P10, P11, P12, P13, P19

Q3 How repeatable is your analysis and negotiation process?
Processes: P5, P6, P8, P9, P10, P13, P17, P19

Q4 How repeatable is your documentation process?
Processes: P6, P8, P9, P10, P13, P14, P15, P16, P19

Q5 How repeatable is your validation process?
Processes: P6, P8, P10, P13, P18, P19, P20

PROCESS

- P1: Follow a written organizational policy for managing the system requirements allocated to the software project (e.g. requirements are documented following a structured standard)
- P2: Establish project responsibility for analysing the system requirements and allocating them to hardware, software, and other system components
- P3: Implement training programme to recognise and meet technical and organisational requirements needs within the project
- P4: Establish process to identify stakeholders in the requirements phase of the project
- P5: Provide adequate resources and funding for managing the allocated requirements in the project (e.g. time, budget, people, tools)
- P6: Establish process to identify skills needs within project, e.g. UML, formal methods, good communication
- P7: Institute process to maintain organisational stability within project, e.g. control staff change
- P8: Explore alternative solutions, requirements techniques and tools for the project
- P9: Establish / maintain process to reach agreement with customer on requirements for project
- P10: Establish/maintain process to involve key stakeholders within the project
- P11: Set realistic goals to address business requirements and requirement process improvement needs within the project
- P12: Establish/implement process to assess feasibility & external environment of project
- P13: Establish/maintain repeatable requirement traceability process that is project-based
- P14: Establish a repeatable process to manage complex requirements at project level
- P15: Establish a repeatable process to manage vague requirements at project level
- P16: Establish a repeatable process to manage requirements growth at project level
- P17: Establish a repeatable process to manage user understanding at project level
- P18: Monitor progress of the set requirements goals
- P19: Agree and document technical and organisational practices specific to project
- P20: Establish a process to review allocated requirements within the project to include software managers and other affected groups

Key:
P = Process
Q = Question

Document 3: Level 2 processes viewed in requirements phases

The questions represent the following 5 requirements phases:

Q1. Requirements Management;
Q2. Requirements Elicitation;
Q3. Requirements Analysis and Negotiation;
Q4. Requirements Documentation;
Q5. Requirements Validation.

For definitions of the processes (P1-P20) and definitions of the 5 requirements phases, please refer to our Web-site at:

http://homepages.feis.herts.ac.uk/~pppgroup/requirements_cmm.htm

The five tables below show each of the requirements phases separately together with their sets of processes. The processes (P1-P20) have been tailored to correspond to each of the five phases.

Q1	How repeatable is your requirements 'management' process?
P1	**Follow a written organisational policy** for managing the system requirements allocated to the software project (to include elicitation, analysis and negotiation, documentation, modelling, verification phases)
P2	Establish project **responsibility** for analysing the system requirements and allocating them to hardware, software, and other system components
P3	Implement **training** programme to recognise and meet technical and organisational requirements needs within the project
P4	Establish process to **identify stakeholders** in the requirements phase of the project
P5	Provide adequate **resources and funding** for managing the allocated requirements in the project (e.g. time, budget, people, tools)
P7	Institute process to maintain **organisational stability** within project, e.g. control staff change

Q2	How repeatable is your 'elicitation' process?
P6	Establish process to **identify skills needs** within *elicitation phase* of the project, e.g. UML, formal methods
P8	Explore **alternative solutions**, requirements techniques and tools for *the elicitation phase* of project
P10	Establish/maintain process to **involve key stakeholders** in requirements *elicitation phase* of project
P11	**Set realistic goals** to address business requirements and requirements process improvements needs within the project
P12	Establish/implement process to **assess feasibility & external environment** of project
P13	Establish/maintain repeatable requirement **traceability process** that is specific to the project
P19	**Define and document** the requirements *elicitation process* (technical and organisational practices specific to project)

Q3	How repeatable is your 'analysis and negotiation' process?
P5	Check adequate **resources and funding** is available for realising the requirements (are they realistic?)
P6	Establish process to **identify skills needs** within *analysis and negotiation phase* of the project, e.g. good communication skills

P8 Explore **alternative solutions,** requirements techniques and tools for *the analysis and negotiation* phase of project
P9 Establish/maintain process to **reach agreement with customer** on requirements for project - to include prioritisation of requirements
P10 Establish/maintain process to **involve key stakeholders** in requirements *analysis and negotiation* phase of project
P13 Establish/maintain repeatable requirement **traceability process** that is specific to the project
P17 Establish a repeatable process to **manage user understanding** at project level
P19 **Define and document** the requirements *analysis and negotiation process* (technical and organisational practices specific to project)

Q4 How repeatable is your 'documentation' process?

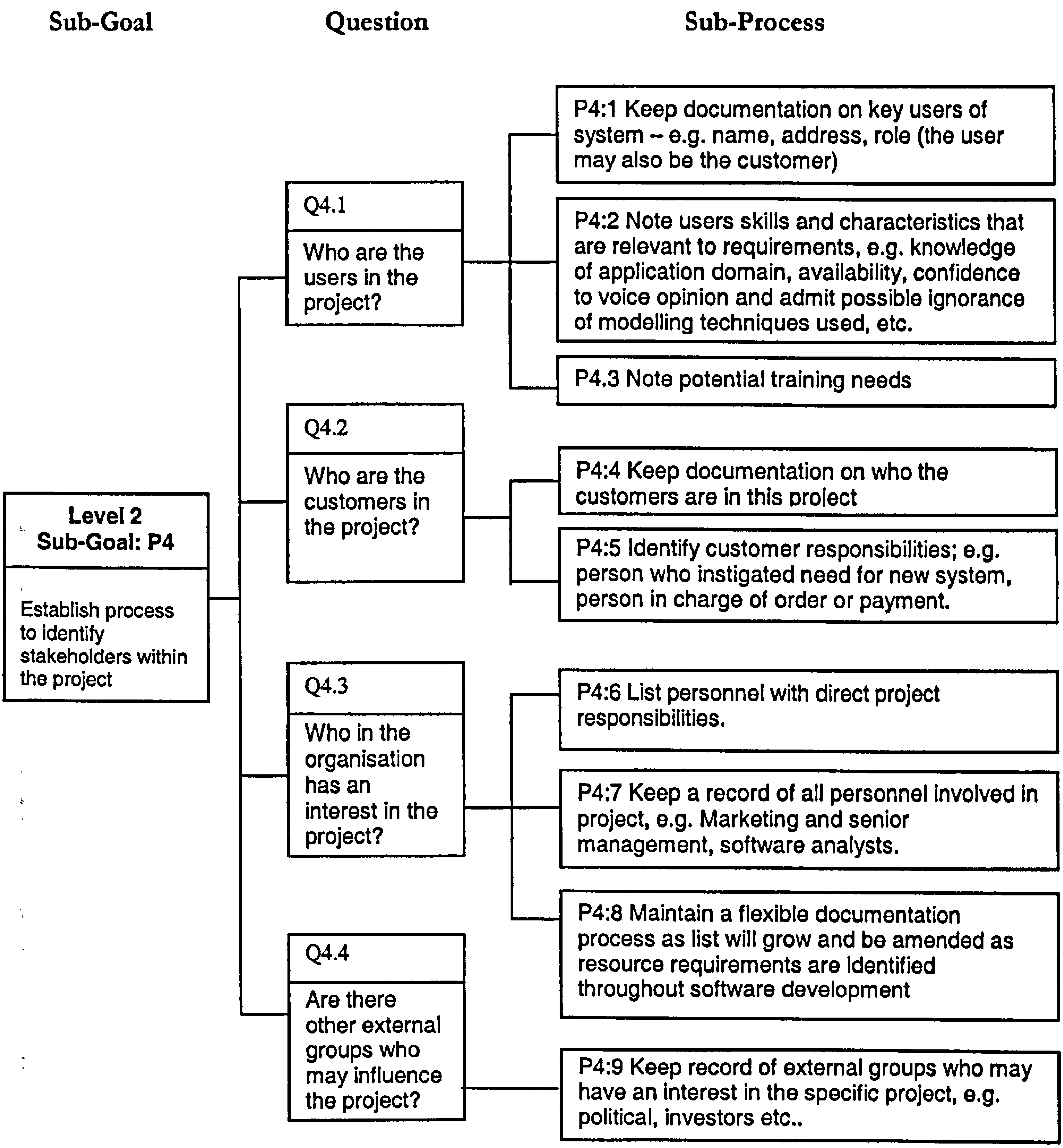
P6 Establish process to **identify skills needs** within *documentation phase* of the project, e.g. modelling skills; ability to use appropriate tools
P8 Explore **alternative solutions,** requirements techniques and tools for the *documentation phase* of project
P9 Establish/maintain process to document **agreement with customer** on requirements for project - to include prioritisation of requirements
P10 Establish/maintain process that *documents* **key stakeholder involvement** in requirements in project
P13 Establish/maintain and *document* repeatable requirement **traceability process** that is specific to the project (e.g. how system requirements develop into software requirements; where requirements are reused)
P14 Establish a repeatable process to **manage complex requirements** at project level
P15 Establish a repeatable process to **manage vague requirements** at project level
P16 Establish a repeatable process to **manage requirements growth/change** at project level
P19 **Define and document** the requirements *documentation* process (technical and organisational practices specific to project)

Q5 How repeatable is your 'validation' process?

P6 Establish process to **identify skills** needs within *validation phase* of the project, e.g. multi-disciplinary; technical writing
P8 Explore **alternative solutions,** requirements techniques and tools for the *validation phase* of project
P10 Maintain process to **involve key stakeholders** in requirements *validation phase* of project
P13 Maintain repeatable requirement **traceability process** that is specific to the project
P18 Monitor progress of the set requirements goals
P19 **Define and document** the requirements *validation process* (technical and organisational practices specific to project)
P20 Establish a process to **review allocated requirements** within the project to include software managers and other affected groups

Document 4: Guideline example of a Level 2 Requirements Process

Establishing a repeatable "Identify Stakeholder' process at a project level



Key:

P = Process
Q = Question

Document 5: Requirements Process Assessment: A practical guide to assessing a Level 2 company

Individual requirements activities are evaluated in order to measure how they are perceived in practice. The results of the assessment will indicate whether the approach, deployment and results of each process implementation are satisfactory and reach the required CMM level goal.

An example of how the assessment is made using a Level 2 process:

The scoring (Table 1) is used to assess the capability of individual requirements processes. Here we use P1 as an example process.

Step 1. The three dimensions given in the evaluation matrix in Table 1 are worked through by a key stakeholder who is involved in the requirements phase:

> 1.1 Approach: The stakeholder believes that management have a wide but not complete commitment to implementing process P1. They assess the approach to be 'fair' and score 4.
> 1.2 Deployment: The stakeholder believes that process P1 has been deployed in some parts of the organisation. They assess the deployment of this process as 'marginally qualified' and score 6.
> 1.3 Results: The stakeholder believes several parts of the organisation are producing higher quality requirements since the introduction of process P1. They assess results to be 'fair' and score a 4.

Step 2. The 3 dimension scores of 4, 6, 4 for this process are added together and divided by 3 – and rounded up. A score of '5' for process P1 is ticked in the appropriate box in the evaluation sheet shown in Figure 1.

TABLE 1: Requirements Capability Scoring Matrix

Score	Key Activity evaluation dimensions		
	Approach	Deployment	Results
Poor (0)	• No management recognition of need • No organisational ability • No organisational commitment • Practice not evident	• No part of the organisation uses the practice • No part of the organisation shows interest	• Ineffective
Weak (2)	• Management begins to recognise need • Support items for the practice start to be created • A few parts of organisation are able to implement the practice	• Fragmented use • Inconsistent use • Deployed in some parts of the organisation • Limited to monitoring/verification of use	• Spotty results • Inconsistent results • Some evidence of effectiveness for some parts of the organisation
Fair (4)	• Wide but not complete commitment by management • Road map for practice implementation defined • Several supporting items for the practice in place	• Less fragmented use • Some consistency in use • Deployed in some major parts of the organisation • Monitoring/verification of use for several parts of the organisation	• Consistent and positive results for several parts of the organisation • Inconsistent results for other parts of the organisation
Marginally qualified (6)	• Some management commitment; some management becomes proactive • Practice implementation well under way across parts of the organisation • Supporting items in place	• Deployed in some parts of the organisation • Mostly consistent use across many parts of the organisation • Monitoring/verification of use for many parts of the organisation	• Positive measurable results in most parts of the organisation • Consistently positive results over time across many parts of the organisation
Qualified (8)	• Total management commitment • Majority of management is proactive • Practice established as an integral part of the process • Supporting items encourage and facilitate the use of the practice	• Deployed in almost all parts of the organisation • Consistent use across almost all parts of the organisation • Monitoring/verification of use for almost all parts of the organisation	• Positive measurable results in almost all parts of the organisation • Consistently positive results over time across almost all parts of the organisation
Out-standing (10)	• Management provides zealous leadership and commitment • Organisational excellence in the practice recognised even outside the company	• Pervasive and consistent deployment across all parts of the organisation • Consistent use over time across all parts of the organisation • Monitoring/verification for all parts of the organisation	• Requirements exceeded • Consistently world-class results • Counsel sought by others

Source: Daskalantonakis, M.K., 1994, "Achieving Higher SEI Levels" IEEE Software, Vol 11, Issue 4.

Figure 1: Evaluation sheet

Organisation: ORG_NAME

CMM Level 2 Date:
KRPA: 'Requirements Management' Process Assessment Average Score:

List of Key processes		0	1	2	3	4	5	6	7	8	9	10
P1	Follow a written organizational policy for managing the system requirements allocated to the software project.						✓					
P2	Establish project responsibility for analysing the system requirements and allocating them to hardware, software, and other system components											
P3	Implement training programme to recognise and meet technical and organisational requirements needs within the project											
P4	Establish process to identify stakeholders in the requirements phase of the project											
P5	Provide adequate resources and funding for managing the allocated requirements in the project (e.g. time, budget, people, tools)											
P7	Institute process to maintain organisational stability within project, e.g. control staff change											

Source: Daskalantonakis, M.K., 1994, "Achieving Higher SEI Levels" IEEE Software, Volume 11, Issue 4

Step 3. Repeat the assessment procedure for each process
Each process can be assessed in the same way and added together to gain an overall score for each requirements phase. E.g. Processes P1, P2, P3, P5 and P7 in the evaluation sheet in Figure 1 undergo the same assessment and the results are added together to receive an overall score for the Requirements Management process.

This process can be extended to include all 5 requirements phases, (Q1-Q5), and the sum of the results of all 5 assessments will result in an overall score for the requirements process. The assessment can also be extended to include a number of key stakeholders recognised as having an influence on the requirements stage of software development.

Step 4. Relating the evaluation score results to the CMM:
Scores documented in the evaluation sheet will relate directly to how individuals perceive requirements process strength. An example of how the scores might relate to CMM levels is given in Figure 2:

Figure 2: Relating Evaluation scores to CMM Level capability

Score	CMM Level
Poor /weak	1
Fair	2
Marginally qualified	3
Qualified	4
Outstanding	5

Please note that these relative scores have not been tested and are open to interpretation.

Appendix G: Examples of Contingency Tables drawn up from Focus Group Data

Table 38: Top Level Problems by CMM Level

Company No	CMM Level	Organisational frq'cy	Organisational percentage	Project frq'cy	Project percentage	Lifecycle/Dev frq'cy	Lifecycle/Dev percentage	Total Problems frq'cy	Total Problems percentage
1*	1	67	39%	75	43%	32	18%	174	100%
2	1	70	50%	49	35%	21	15%	140	100%
3	1	43	34%	70	56%	13	10%	126	100%
4	1	70	46%	48	31%	35	23%	153	100%
7	1	6	9%	34	49%	30	43%	70	100%
10	1	21	27%	25	32%	32	41%	78	100%
Level 1 Total	1	277	37%	301	41%	163	22%	741	100%
8	2	37	39%	36	38%	21	22%	94	100%
11	2	15	25%	26	43%	20	33%	61	100%
Level 2 Total	2	52	34%	62	40%	41	26%	155	100%
6*	3	66	50%	42	32%	25	19%	133	100%
9	3	14	42%	10	30%	9	27%	33	100%
13	3	32	29%	55	50%	23	21%	110	100%
Level 3 Total	3	112	40.4%	107	37.3%	57	22.3%	276	100%
5*	4	54	68%	21	27%	4	5%	79	100%
Level 4 Total	4	54	68%	21	27%	4	5%	79	100%
TOTAL PROBLEMS		495		491		265		1251	100%

* Formal CMM assessment undertaken

Table 39: All Companies /all problems by problem group in total frequency order

	Practitioner Groups									Total		
	Dev (19 groups)			*PM (17 Groups)*			*SM (9 Groups)*			*(45 Practitioner Groups)*		
Problems	*Frq*	*% col*	*% row*	*Frq*	*% col*	*% row*	*Frq*	*% col*	*% row*	*Frq*	*% col*	*% row*
Organisational Issues												
People	83	13	50	51	12	31	33	16	20	167	13	100
Communication	72	11	55	38	9	29	21	10	16	131	10	100
Change Management	38	6	45	35	8	42	11	5	13	84	7	100
Culture	22	4	48	14	3	30	10	5	22	46	4	100
Goals	18	3	46	10	2	26	11	5	28	39	3	100
Politics	14	2	50	5	1	18	9	4	32	28	2	100
Org'l Total	**247**	**39**	**50**	**153**	**37**	**31**	**95**	**46**	**19**	**495**	**40**	**100**
Project Issues												
Tools/Technology	82	13	55	49	12	33	19	9	13	150	12	100
Documentation	76	12	53	48	11	34	19	9	13	143	11	100
Budget/Estimates	33	5	34	44	11	45	20	10	21	97	8	100
Timescales	23	4	37	31	7	49	9	4	14	63	5	100
Quality	16	3	42	13	3	34	9	4	24	38	3	100
Project Total	**230**	**37**	**47**	**185**	**44**	**38**	**76**	**37**	**15**	**491**	**39**	**100**
Lifecycle/Dev Process												
Requirements	71	11	55	39	9	30	20	10	15	130	10	100
Testing	39	6	62	14	3	22	10	5	16	63	5	100
Maintenance	18	3	51	15	4	43	2	1	6	35	3	100
Design	17	3	81	4	1	19	0	-	0	21	2	100
Coding	6	1	38	8	2	50	2	1	13	16	1	100
Lifecycle Total	**151**	**24**	**57**	**80**	**19**	**30**	**34**	**17**	**13**	**265**	**21**	**100**
Total	**628**	**100**	**50**	**418**	**100**	**33**	**205**	**100**	**16**	**1251**	**100**	**100**

Appendix H: Processes involved in creating a Correspondence Analysis map of Software Development Problems

An Empirical analysis: Identifying, classifying and analysing SPI problems

1. Focus Group Interviews
Over 200 software practitioners were asked : "What are the obstacles to improving your software processes?"

Interview data creates over 2,000 pages of transcripts

2. A Content Analysis of transcripts resulted in classifying 16 problem groups

To increase confidence in my classification I performed a

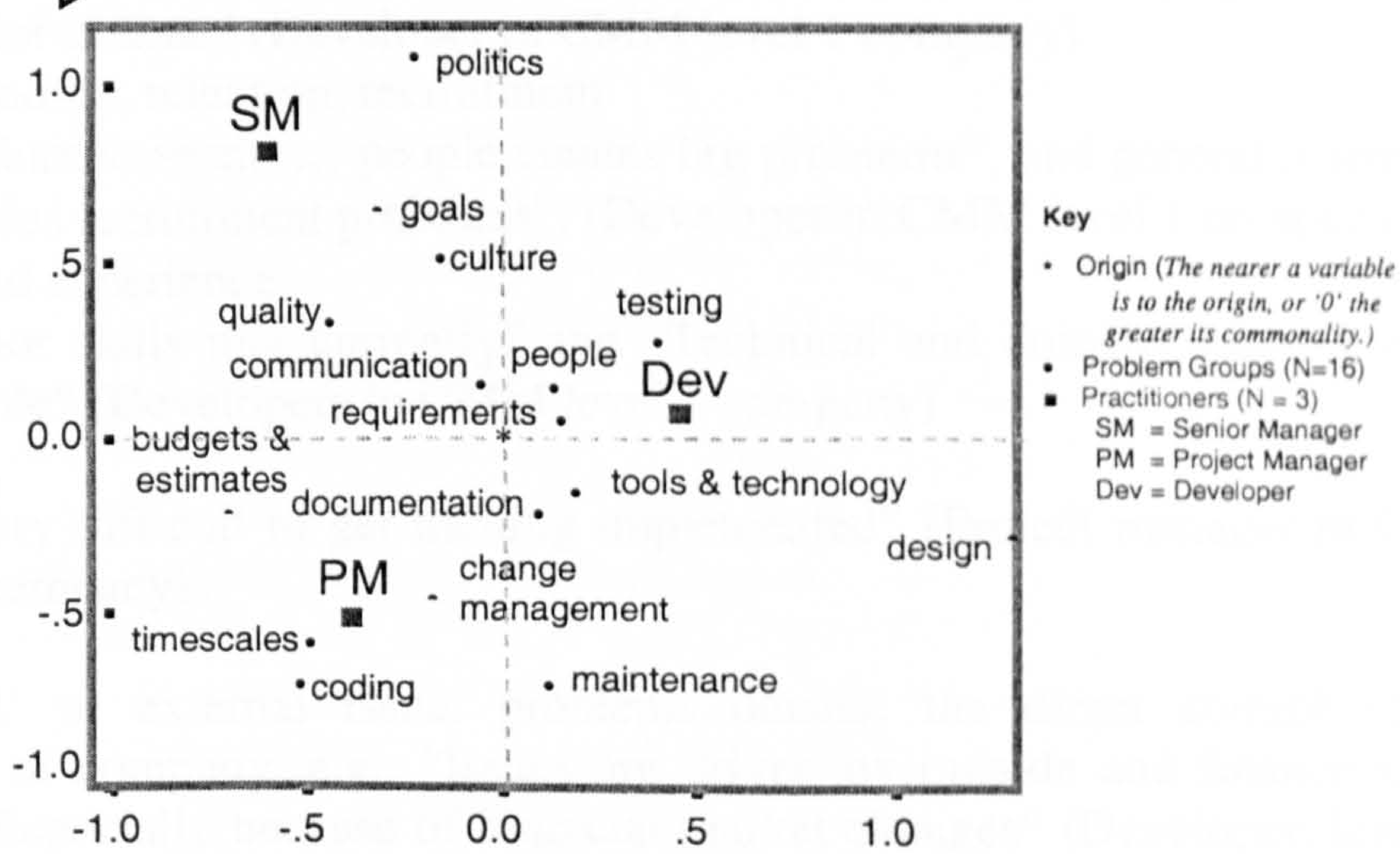
3. Cohen Kappa Inter-rater reliability test. Results show a 'substantial' level of agreement

I drew up contingency tables based on problem frequencies

4. 1,251 problems were placed into problem groups and referenced to 3 practitioner groups: Developers; Project Managers; and Senior Managers

I used profiles from the contingency tables to create a ...

5. Correspondence Analysis (CA) Map
The map graphically displays the relationships between Practitioner and Problem Nominal Variables:



6. Results: The CA Map reveals practitioner group problems through proximity of variables, e.g. Senior Managers (SMs) are associated with problems relating to politics & goals. The most common problems relate to 'people', 'requirements', 'tools and technology', 'documentation' and 'communication' as they are placed near to the origin.

s.beecham@herts.ac.u

EPSRC

University of Hertfordshire

Appendix I: SPI Problem Classification

I have broken down problems that practitioner groups are encountering in their software development into 3 discrete process areas: 'Organisational', 'Software Development Lifecycle' and 'Project'. These classifications were drawn directly from the focus group transcripts – all problems were given equal focus. They were not chosen to link directly to key process areas in the CMM and as a result there are some areas that are outside the scope of the SW-CMM.

Organisational Issues

1. **Communication**
   1.1 Internal (within and between departments)
   E.g. "We don't talk to the modelling department, we don't always talk to hardware department, we don't talk to systems. It is the interfaces, communication interfaces that I think is the biggest problem" (Developer in CMM level 1 company)
   1.2 External (between any internal practitioner or group with external customers/users)
   E.g. "We also suffer from having little to no communication with the users at ground level" (Developer in CMM level 1 company)
   1.3 Physical distance/split sites/geography/company structure
   E.g. "Physical siting of teams is poor. Not under management control" (Senior Manager in CMM level 1 company)
2. **People Issues**
   2.1 Responsibilities, rewards, blame
   E.g. ".. responsibilities are not clear and some 'buck passing' goes on.. can waste a lot of time" (Developer in CMM level 1 company)
   2.2 Staff turnover, retention, recruitment
   E.g. "failure to retain ... people creates big problems", and general resourcing is low, plus recruitment problems". (Developer in CMM level 1 company)
   2.3 Skills and experience
   E.g. "Poor skills mix currently" and "Technical and domain over-reliant on key people" (Developers in CMM level 1 company)
   2.4 Training
   E.g. "Very difficult to get training implemented" (Project manager in CMM level 1 company).
3. **Politics**
   A 'political' or external issue: problems outside the direct control of the practitioner or company, e.g., "Issues are driven by outside and finance senior managers. Especially because of financial market changes" (Developer, level 1). "Since the take-over, cost and benefit issues move to the forefront". (Senior Manager, CMM level 1). "Software Managers are now having to deal with political, commercial issues externally" (Senior Manager, CMM level 4 company).
4. **Culture**
   This category relates to ingrained behaviour, ways of thinking or habit specific to a group or company. It includes problems with cultivating a desirable company culture and problems with the existing culture. E.g. "There are very big cultural differences between here and other {sites} and we are very much driven by USA

culture. Sometimes you see things come in that might be good for the particular country it came from but it may not necessarily work here" (Developer, CMM level 4 company).

4. **Goals**
   This category relates to problems with company goals and objectives. This includes setting, keeping, attaining, identifying, and communicating 'goals'. E.g., "It suddenly became a point that you had to do something in [the department] as one of your *goals* for the year and if you didn't you were a naughty person, with the result that people resented being forced into something", (Developer, CMM level 4 company).

5. **Change Management**
   This category encapsulates how companies are coping with change and reorganisation at any CMM level, e.g., "It is very difficult to show the benefits of change to people", (Developer in level 1 company). "Middle management tend to be a difficult area to adopt change, they tend to moan a lot about a need for change. At a senior level, you have a strong desire to improve the process, but in between there seems to be more resistance to implement change" (Senior Manager in CMM Level 1 company).

Software Development Life Cycle Issues

Companies vary in their use of software lifecycles. Whichever form is used (e.g. waterfall, spiral, single prototype), the processes inmy classification scheme are likely to appear:

1. **Requirements**
   Elicitation, specification/modelling, verification. Requirements problems are identified as complex; vague/undefined; subject to growth/creep; poor user understanding; incomplete; lacking in traceability, inflexible. For example, "Requirements is a big problem. It is difficult to get any in the beginning ..."; "Requirements are very vague"; "Interpretation of requirements is very difficult. There are gaps in them. Not detailed enough"; *and conversely,* "Some customers define things in such detail that everything is tied down far too tightly."
2. **Design** - correct, understandable, flexible, methods, For example, "designs have got so much larger and the procedures got left behind because you have skilled people who are used to writing small amounts of code who got caught in the line"; "[in] one of the project teams ... there was no normal design done".
3. **Coding** – consistent, documented/comments, traceable; reusable. "software has got bigger far quicker in the last 5/6 years. The number of lines of code you write went up from 5,000/6,000 to say 100,000/200,000".
4. **Testing** – scalable; measurable; reliable, for example, "I think that is an area well worth improving, testing. I mean automate it, really, if we can. But it is a very expensive activity, very expensive"; "I think it would be better if we had ... a separate test function developed".
5. **Maintenance** – patches, updates, bugs; defects; regression; release; reuse; fault fixing. For example, "[we need to] reduce the backend cost, which we all know is where the biggest chunk of the money goes to fix the problem." "I think one of

our biggest problems is that we have still got a number of legacy projects. ... And these old projects don't take very well to having their processes upgraded to reflect the present. And I think that is what is holding us back".

Project Issues

I have identified project-related issues as:

1. **Budget and estimates**
   Investment and resources, lack of funding for projects. I look at direct causes of problems; e.g. "we don't have enough money to invest in new systems" comes under budget, as it is a problem with investment. I can't tell that a new system would help, all I know is that they cannot even test the possibility due to a lack of funding.
   Resources can cover finance, personnel and equipment.
2. **Documentation** Includes measurement data; written procedures; and problems of
   a) co-ordination and management of documentation
   b) feedback and post-mortems on documentation
   c) data collection methods
3. **Quality**
   Includes control problems and the tension between producing high quality products within given timescales and how quality impacts other areas of software development. High quality requirements can take resources from other areas e.g. SPI.
4. **Timescales**
   Commitment to complete project within a certain time places pressure on developers and project managers. Problems with timescales in projects impact SPI and quality. Inaccurate estimates for project timescales can place pressure on developers. Tension between sales and developers.
5. **Tools and technology**
   Includes implementation of tools and new initiatives. SPI is seen as a tool for improving software. E.g. "A SPI initiative started in a company some time ago got shot down in flames", is seen as a tools and technology problem. The category includes productivity/volume of work/pressure, e.g. "we have a problem keeping up to date with the generation of languages" (we cannot tell whether this is an investment or a training issue – all we know is that they are having a problem with technology).

Appendix J: Requirements Engineering Problem Classification

ORGANISATIONAL ISSUES

Culture: Ingrained behaviour, ways of thinking or habit specific to a group or company. It includes problems with cultivating a company culture that recognises and deals with requirements problems. Social aspects concerned with organisational change and organisational politics.

Developer Communication: Communication between staff groups within the Company is often poor. E.g. Marketing will make promises to customers that cannot be met by Software Group, or Requirements Engineers do not communicate adequately with Software group about feasibility of design.

Resources: This relates to time, costs, investment in requirements tools and people. Projects are not keeping to time –unreliable estimates being given at beginning of project/ management reluctant to provide extra resources (staff time/training/costs of new tools) towards improvement activities as they are generally looking at the short term.

Skills: Over dependence on few experienced staff. Not sharing of best practice.

Staff retention: This incorporates recruitment and workforce stability. Companies are having difficulties in recruiting staff of the right level and retaining experienced requirements staff.

Training: Requirements training needs are not being met.

User Communication: Difficulties the company is having in communicating with outside users (e.g. inflexible company structure dictates who should discuss customer requirement needs with the customer – often precluding software designers).

REQUIREMENTS ENGINEERING PROCESS/TECHNICAL ISSUES

Complexity of application: Problems inherent in large-scale projects that can span many years and sites: can be highly complex, may need to be highly reliable, safety critical and customized.

Poor User Understanding: User understanding of personal needs is often confused and undetected until too late. E.g., a customer will often ask for functions that are not needed and prove difficult to implement.

Requirements Growth: Lack of control over constraints, growth of requirements, requirements changes over time

Requirements traceability: A link or definable relationship between entities where a lack of traceability can lead to lost work and creates difficulties when sharing work across teams.

Undefined Requirements Process: No documented methods for undertaking requirements related activities. Lack of a defined requirements process can result in a chaotic system.

Vague requirements: Also called 'Tacit Requirements'. Here requirements capture or software requirements specification/documentation is incomplete, ambiguous or flawed.

Appendix K: RE Process Problem Tables

Table 40: Requirements Organisational Issues

REQUIREMENTS ORG ISSUES (12 companies, 45 focus groups)	CMM Level 1 6 co's	CMM Level 2 2 co's	CMM Level 3 3 co's	CMM Level 4 1 co.	Total
1. Culture/procedures	4	3	1	10	18
2. Developer communication	31	6	17	1	55
3. Resources	26	3	5	0	34
4. Skills and Responsibilities	29	6	9	2	46
5. Staff retention/ recruitment	21	6	1	1	29
6. Training needs not met	15	3	1	1	20
7. User Communication	13	5	12	0	30
Total – Observed and (normalised)	139 (139)	32 (96)	46 (92)	15 (90)	232 (417)

Table 41: Requirements Technical Issues

REQUIREMENTS TECHNICAL ISSUES (12 companies, 45 focus groups)	CMM Level 1 6 co's	CMM Level 2 2 co's	CMM Level 3 3 co's	CMM Level 4 1 co.	Total
8. Complexity of application	8	8	11	0	27
9. Requirements traceability	4	0	0	0	4
10. Poor user understanding	2	1	2	0	5
11. Requirements growth	14	7	9	1	31
12. Undefined requirements process	21	6	5	0	32
13. Vague initial requirements	24	5	4	0	33
Total - Observed and (normalised)	73 (73)	27 (81)	31 (63)	1 (6)	132 (223)

Table 42. Organisational problems across companies

	Co.1 CMM Level 1*		Co.2 CMM Level 1+		Co.3 CMM Level 1+		Co4 CMM Level 1+		Co.5 CMM Level 4*		Co.6 CMM Level 3*		Co.7 CMM Level 1+	
	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%
Developer communication	11	37	3	14	8	36	6	15	1	7	7	41	2	25
Skills & Responsibilities	4	13	8	36	2	9	13	32	2	13	0	0	0	0
Resources	7	23	0	0	3	14	8	20	0	0	4	24	4	50
Staff retention/ recruitment	3	10	5	23	4	18	6	15	1	7	1	6	0	0
User communication	2	7	4	18	0	0	2	5	0	0	5	29	1	13
Training	1	3	2	9	5	23	5	12	1	7	0	0	1	13
Culture	2	7	0	0	0	0	1	2	10	67	0	0	0	0
Total Organisational Problems	30	100	22	100	22	100	41	100	15	100	17	100	8	100

	Co.8 CMM Level 2+		Co.9 CMM Level 3+		Co. 10 CMM Level 1+		Co 11 CMM Level 2+		Co 13 CMM Level 3+		Total No of Problems	
	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%
Developer communication	5	23	0	0	1	6	1	10	10	42	55	24
Skills & Responsibilities	4	18	2	40	2	13	2	20	7	29	46	20
Resources	1	5	0	0	4	25	2	20	1	4	34	15
Staff retention/ recruitment	6	27	0	0	3	19	0	0	0	0	29	13
User communication	2	9	3	60	4	25	3	30	4	17	30	13
Training	2	9	0	0	1	6	1	10	1	4	20	9
Culture	2	9	0	0	1	6	1	10	1	4	18	8
Total Organisational Problems	22	100	5	100	16	100	10	100	24	100	232	100

* indicates formal CMM assessment

+ indicates self assessment

Table 43. Requirements Technical problems across companies

	Co.1 CMM Level 1*		Co.2 CMM Level 1+		Co.3 CMM Level 1+		Co4 CMM Level 1+		Co.5 CMM Level 4*		Co.6 CMM Level 3*		Co.7 CMM Level 1+		Co.8 CMM Level 2+		Co.9 CMM Level 3+		Co. 10 CMM Level 1+		Co 11 CMM Level 2+		Co 13 CMM Level 3+		Total No of Problems	
	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%	Frq	%
Vague initial reqs	4	22	1	33	1	20	8	31	0	0	1	11	3	50	1	11	1	13	7	47	4	22	2	14	33	25
Undefined reqs process	9	50	0	0	0	0	9	35	0	0	2	22	2	33	1	11	0	0	1	7	5	28	3	21	32	24
Requirements growth	4	22	0	0	3	60	5	19	1	100	2	22	0	0	2	22	3	38	2	13	5	28	4	29	31	23
Complexity of application	0	0	0	0	0	0	3	12	0	0	4	44	1	17	5	56	2	25	4	27	3	17	5	36	27	20
Poor user understanding	0	0	1	33	0	0	1	4	0	0	0	0	0	0	0	0	2	25	0	0	1	6	0	0	5	3.8
Requirements Traceability	1	5.6	1	33	1	20	0	0	0	0	0	0	0	0	0	0	0	0	1	7	0	0	0	0	4	3
Total Technical Problems	18	100	3	100	5	100	26	100	1	100	9	100	6	100	9	100	8	100	15	100	18	100	14	100	132	100

* indicates formal CMM assessment

+ indicates self assessment

Table 44: Technical Requirements Problems by CMM Group and Practitioner Group										
	CMM Level 1		*CMM Level 2*		*CMM Level 3*		*CMM Level 4*		*Total No of Problems*	
	6 co's	*Norm*	*2 co's*	*Norm*	*3 co's*	*Norm*	*1 co*	*Norm*		
	Frq	*x 1*	*Frq*	*x 3*	*Frq*	*x 2*	*Frq*	*x 6*	*Frq*	*Norm.*
Requirements growth/change										
SM (9 focus groups)	1	1	0	0	2	4	0	0	3	5
PM (17 focus groups)	3	3	2	6	0	0	1	6	6	15
Dev (19 focus groups)	10	10	5	15	7	14	0	0	22	39
Total	14	14	7	21	9	18	1	6	31	59
Vague initial requirements										
SM (9 focus groups)	4	4	1	3	1	2	0	0	6	9
PM (17 focus groups)	8	8	1	3	0	0	0	0	9	11
Dev (19 focus groups)	12	12	3	9	3	6	0	0	18	27
Total	24	24	5	15	4	8	0	0	33	47
Undefined requirements process										
SM (9 focus groups)	2	2	0	0	0	0	0	0	2	2
PM (17 focus groups)	8	8	3	9	0	0	0	0	11	17
Dev (19 focus groups)	11	11	3	9	5	10	0	0	19	30
Total	21	21	6	18	5	10	0	0	32	49
Poor user understanding										
SM (9 focus groups)	0	0	0	0	0	0	0	0	0	0
PM (17 focus groups)	1	1	1	3	0	0	0	0	2	4
Dev (19 focus groups)	1	1	0	0	2	4	0	0	3	5
Total	2	2	1	3	2	4	0	0	5	9
Inadequate Req's Traceability										
SM (9 focus groups)	0	0	0	0	0	0	0	0	0	0
PM (17 focus groups)	3	3	0	0	0	0	0	0	3	3
Dev (19 focus groups)	1	1	0	0	0	0	0	0	1	1
Total	4	4	0	0	0	0	0	0	4	4
Complexity of application										
SM (9 focus groups)	0	0	4	12	2	4	0	0	6	16
PM (17 focus groups)	4	4	2	6	3	6	0	0	9	16
Dev (19 focus groups)	4	4	2	6	6	12	0	0	12	22
Total	8	8	8	24	11	22	0	0	27	54
Total number of problems internal to the requirements process										
SM	7	7	5	15	5	10	0	0	17	32
PM	27	27	9	27	3	6	1	6	40	66
Dev	39	39	13	39	23	46	0	0	75	124
Total Technical Problems	73	73	27	81	31	62	1	6	132	222
Normalising the Focus Groups gives:										
SM 9 groups (x 1.7)	12		9		9		0		29	
PM 17 groups (x .9)	24		8		3		1		36	
Dev 19 groups (x .8)	31		10		18		0		60	

Appendix L Defining Requirements Processes at Level 2 maturity

P1: Follow a written organisational policy for managing the system requirements allocated to the software project

This process is taken directly from the SW- CMM: Requirements Management, Key Process Area, Commitment to Perform, Commitment 1 – (Paulk, 1995).

Literature in support of this process includes, e.g. (Sommerville and Sawyer 1997) p.223; (Cugola and Ghezzi 1998); (Sawyer *et al.* 1997 4.4); (Pfleeger and Rombach 1994);(Fayad 1997); (Christie 1999)

This process is broken down as follows

- Each <requirements> activity is performed "according to a documented procedure". (CMM Template, Activity 2, Paulk, 1995, p.45)
- The written policy will define processes in requirements activities (CMM SPP Activity 5 (Paulk *et al.* 1995).
- The written policy will document process goals.
- The written policy will serve to include people who have a central role in performing the activities needed to accomplish the process goals. The definition must reflect and support the need for "co-operation among people" and "must be highly flexible" (Cugola and Ghezzi 1998) (Sawyer *et al.* 1997 4.4).

Sommerville and Sawyer (1997) recommend that the management of allocated requirements should include the following policies:

1. a set of objectives for [the requirements management] process and rationale associated with each of these objectives
2. the reports to make the requirements engineering process visible and the activities which are expected to produce these reports as deliverables
3. the standards for requirements documents and requirements descriptions which should be used
4. change management and control policies for requirements
5. requirements review and validation policies
6. relationships between requirements management and other system engineering and project planning activities
7. traceability policies which define what information on dependencies between requirements should be maintained and how this information should be used and managed.
8. Criteria when these policies can be ignored; in these situations, managers use their own judgement on how to implement a requirements change.

Sommerville and Sawyer (1997) place this management activity in their list of basic guidelines for their Level 2 companies in their requirements engineering good practice guide, stating:

> *"Requirements management policies define goals for requirements management, the procedures which should be followed and the standards which should be*

used. These policies should be explicitly defined as part of your quality management system. ... Explicit policies tell people involved in the process what they are expected to do and why it should be done... Projects generally manage their requirements in comparable ways, so with explicit policies, there is less dependence on individual knowledge and expertise.

In order to define policies, you must understand your existing processes for requirements management. This is likely to reveal problem areas which may become the focus of process improvements.

(Sommerville and Sawyer 1997) p.223 .

P2: Establish project responsibility for analysing the system requirements and allocating them to hardware, software, and other system components.

This process is taken directly from the Software CMM (Paulk *et al.* 1995) Requirements Management (RM) Key Process Area (KPA) Ability 1. The CMM emphasises within each KPA the need to establish responsibility for project tasks, e.g. RM, Ability to Perform 1. "Analysis and allocation of the system requirements is not the responsibility of the software engineering group but is a prerequisite for their work".

This is also a main section in (McFeeley 1996) p.98, section 3.8 "Finalize Roles and Responsibilities of the Various Infrastructure Entities. (Scharer 1990) Practitioners should take responsibility.

P3: Implement training programme to recognise and meet technical and organisational requirements needs within the project.

"A lack of training.. led to teams that were less familiar with the RE process (Hofmann and Lehner 2001)

The training programme should provide a platform for explaining why the organization is spending time and effort on a Requirements Process Improvement program. As practitioners' understanding grows so will their support. They must be motivated to join in the effort and assist it. The motivation should address the following points:

- Why change?
- What's wrong with the status quo?
- Why should I care?
- When will I be affected (immediately or sometime in the future)?

For further information on motivating practitioners in software process improvements efforts see (McFeeley 1996) section 3.6; and (Baddoo and Hall 2002).

The ami guide also emphasises the need for 'properly administered' training, stating that "an assessment of the different needs and levels of training has to be made" (ami 1992).

In his section on the Team Software Process (TSP), Humphrey (Humphrey 2002) states that "the biggest single problem with the TSP is training. With few exceptions, mangers want the benefits .. but are reluctant to invest in the required training. [Using the improvement method] with untrained or partially trained teams .. have always failed". Humphrey recommends that organizations implement his improvement program properly or not even try it. This could be applied to the CMM too.

P4: Establish process to identify stakeholders within the requirements phase of the project

Stakeholder identification is not explicitly modelled in the Software-CMM, yet it is one of the most critical processes in terms of practitioner feedback and problems cited in the literature.

(Paulk *et al.* 1995) explain how the CMM addresses the customer

> 'The CMM is written from a software perspective. It covers the software process and addresses only those requirements allocated to software. It does not cover the processes of the customer or the system engineering group. It does describe inter-group interfaces that the software engineering group should proactively address, hopefully in a spirit of teamwork and an effective customer-supplier relationship." pp 53-54.

This description of the CMM shows that inter group processes involving customers (and users) and the system engineering group are implicit rather than explicit. My empirical research details developer communication and user communication problems as accounting for 24% and 12% (total 36%) of organisational-based requirements problems (Hall *et al.* 2002). I interpret this as the stakeholder (to include customer and system engineering group) process being poorly defined and implemented.

Stakeholder identification is also central to Sommerville and Sawyer's (1997) Practical Process Improvement Guidelines: "The stakeholders in a system should always be explicitly identified in the requirements document and if appropriate information should be maintained which links specific requirements to the stakeholder who proposed these requirements" P.73.

A survey carried out by Barry Boehm and his team in their 'EasyWinWin' project asked practitioners the question "What are your major concerns with your organization's typical requirements approach? 5 concerns were mentioned, of which "Key stakeholders are excluded" was a major concern (Boehm 2001). The Standish Group's Chaos report (StandishGroup 1995) also identified "lack of user input" as contributing to 12.8% of project failure. And, lastly, Dorfman in (Thayer and Dorfman 1990) states that good requirements include an "agreement among developers, customers, and users on the job to be done and the acceptance criteria for the delivered system" p.4.

In a case study for improving RE, the author find the biggest problems were associated with communicating with internal customers who did not participate in the introduction of the new methods (Jacobs 1999). Further literature in support of identifying stakeholders include:(Hofmann and Lehner 2001) and users (El Emam et al. 1996).

P5: Provide adequate resources and funding for managing the allocated requirements in the project

This process forms a part of the SW-CMM that demonstrates an 'ability' to perform the requirements activities: (Paulk et al, 1995), Requirements Management Key process area (Commitment 1; Ability 3).

The requirements process is a microcosm of the software process and as such organisations need to "Launch the [SPI] program by building an understanding and an awareness of the costs and benefits" and "Commit the resources necessary" (McFeeley 1996).

Not only does the requirements process need resources to perform the activities, part of its activities is to provide "A good basis for resource estimation (cost, personnel quality and skills, equipment and time) Dorfman, in (Thayer and Dorfman 1990).

P6: Establish process to identify skills needs within the project (for example, the skills required in requirements elicitation)

This requires matching the needs of project to the skills of personnel (Hofmann and Lehner 2001) It is not a process found in the Software CMM, but is included in the PEOPLE CMM Level 2: Skills (Curtis *et al.* 1995).
There is a general discussion on personnel and the sensitive issue as to how to rate personnel capability and personnel experience in (Boehm 1981)

(El Emam and Madhavji 1995) in their field study have a section on skills sets in their field study. They recommend that appropriately skilled people be assigned to analyst and architect positions, as well as skilled users in the requirements process especially the principal user - project managers should also have a high capability in the requirements engineering phase.

P7: Institute process to maintain stability within project, e.g. cope with changes in staff/ requirements priorities/general priorities in organising the requirements process.

"A disciplined software engineering process helps address many 'accidental' difficulties" S. Faulk, Software requirements: A Tutorial" in (Dorfman and Thayer 1997) "To achieve a stable project over a long period of time, a manager must encourage the project to function .. with a fresh supply of trainees coming one end and a stream of experienced leaders coming out of the other" ... and "A project is not a house of cards which collapses when a single key person is removed .. when management thinks it is, the prophecy becomes self-fulfilling" "If a [practitioner] is indispensable, get rid of him as quickly as possible"!! all quotes from Chapter "Stability through change", in (Weinberg 1998).

Recognise and anticipate volatile requirements: e.g. mutable requirements; emergent requirements; consequential requirements and compatibility requirements (see (Kotonya and Sommerville 1998) p.116).

Successful RE teams manage requirements priorities "To specify prioritized requirements, the RE team develops various models together with prototypes" (Hofmann and Lehner 2001)
McFeeley has a section dedicated to prioritizing activities and developing an improvement agenda (McFeeley 1996).
"The baselines, particularly the maturity baseline, typically identify issues and provide recommendations based on a much broader consensus than may have been available before.
These issues and recommendations serve to provide some guidance, and often, a prioritization of actions."
(A Level 2 organisation should be in a position to identify where their priorities lie as they must have their baseline maturity processes in place).
Another guide to creating a stable environment is found in (McFeeley 1996), where McFeeley advocates that organisations "Establish Software Process Improvement Infrastructure"
in order to "build the mechanisms necessary to help the organization institutionalize continuous process improvement. .... A solid, effective infrastructure can sustain a developing [SPI] program until it begins to produce visible results. Unsupported [SPI] programs can become isolated and die out during periods of stress and tension within their organizations.... To effectively manage the SPI program, an infrastructure must be in place or created."
Coad puts forward Object Oriented Analysis techniques as a method for managing continual change in requirements (Coad and Yourdon 1990)

P8: Explore alternative solutions, requirements techniques and tools for the project

"Several methods and languages can be used for specifying the functionality of computer systems. No single language, of those now available, is equally appropriate for all methods, application domains, and aspects of a system. Thus users of formal specification techniques need to understand the strength and weaknesses of different methods and languages before deciding on which to adopt.

A review of formal methods: Robert Vienneau, in (Dorfman and Thayer 1997)

"We expect methods to be panaceas – medicines that cure all diseases. This cannot be."(Jackson 1995) Classifying problems and relating them to suitable methods is a central theme of Jackson's (1995) book.

There is not a one size fits all technique, and in a study of three different projects (Lauesen and Vinter 2001) conclude "the value of a technique depends on the project".

In a study of management of process improvement by prescription (Middleton and McCollum 2001) conclude that "the idea of a 'best' method is misleading because of the diverse range of projects and developers". The generic lesson gleaned for their research is that an organization is "probably unwise to use a heavily prescriptive methodology to improve its software development performance" (Middleton and McCollum 2001).

In the documentation stage it may be necessary to use well defined semantics, such as deterministic finite state machines, Petri nets, decision trees, propositional calculus, predicate calculus to avoid ambiguity... the choice will be driven primarily by expressive power and suitability for the aspect of the system." (Davis *et al.* 1993). However, Davis does admit that replacing natural language with formal notations greatly decreases ambiguity in the SRS but almost always at the expense of understandability (except for decision trees). He therefore suggests augmenting natural language with more formal models.

Other recommendations include:

Requirements should be 'explored' through methods such as: brainstorming, simulation, visualization, storyboard illustrations and scenarios (Maiden and Gizikis 2001).

Measurement techniques are used to help explore and understand the size of the product and manage project constraints such as duration, time-to-market and productivity, along with customer satisfaction factors e.g. MkII Function Point Analysis (Rule 2001) – Function point analysis is used to measure productivity of system development and system maintenance, and can also be used for project estimating by converting function points into work-effort (Onvlee 1995).

In a case study by Kitchenham (Kitchenham 1995) function points are said to be flawed - don't give accurate predictions of effort, are over-complex as metrics and are unsuitable for cross-company comparisons – signalling that organisations must be cautious about the methods they use and the results they obtain. Yet, in another case study, function point analysis gives slightly better results for effort prediction than using the COCOMO model (Boehm 1981) and (Stricker 1995) states that their model

(F-PROM) brings better results than either function point analysis or the COCOMO model.

The general message is, understand the technique you are using, acknowledge its strengths and weaknesses and assess whether there may be a better way of achieving your aims.

P9: Establish/maintain process to reach agreement with customer on requirements for project.

This processes does not form part of the Software CMM Requirements Management Key Process Area activities, but is included in its definition (Paulk et al, 1995).

Agreement includes "Obtain Approval for [SPI Proposal] and Initial Resources" (McFeeley 1996) section 1.5.

Consider ethnographic solutions as presented in viewpoints (Hughes and 27-34 1995)

"...good requirements include .. Agreement among developers, customers, and users on the job to be done and the acceptance criteria for the delivered system." Dorfman in (Thayer and Dorfman 1990)

P10: Establish/maintain process to involve key stakeholders within the project.

"Involving stakeholders early .. resulted in an increased understanding of the RE process being used" and "Requirements prioritized by stakeholders drive successful RE teams "(Hofmann and Lehner 2001). There is a need to develop a trust and a shared vision of what the project is trying to achieve; "users are part of the system and therefore it is necessary that their capabilities are explicitly grown with the system..." (Middleton and McCollum 2001). User contribution should include involvement, expression, participation and commitment (Middleton and McCollum 2001). "The seeds of failure are often sown at this point in the requirements elicitation process. Many organizations lack the ability to consolidate and reconcile multiple stakeholder viewpoints (Cottengim 2002).

(Paulk *et al.* 1995) pp 53-54 indicate that there is nothing explicit in CMM. Yet heavily supported in my collaborative research (Hall *et al.* 2002). And the literature, e.g. (Hofmann and Lehner 2001) (Sommerville and Sawyer 1997) p.73; (Boehm 2001);(StandishGroup 1995) (Thayer and Dorfman 1990)

"Analysts need to communicate throughout the analysis effort, the must communicate just to extract the problems space and requirements from the client ..." (Coad and Yourdon 1990)

"Users should always participate in the requirements engineering process" (El Emam and Madhavji 1995)

P11: Set realistic improvement goals to address problems in the requirements process project

The process of setting realistic goals is important for

1) modelling the right level of 'project' improvement goals for the requirements phase to solve recognised problems, and
2) in setting functional and non-functional 'requirements'.

"When there is a perception that the requirements are unrealistic, software developers may become discouraged and not fully commit tot he goals of the project" (Linberg 1999)

"Determine Key Business Issues Purpose: Unless the SPI program is driven by the current business needs and understood and agreed to by management, it will likely be difficult to sustain the program over the long haul. This is because it will be difficult to clearly demonstrate to senior management that the initiative is achieving real value for the organization in business terms" (McFeeley 1996).

"For any process model to be effective in the specific project in hand, there is a need to customise the model according to the project goals. This may be achieved by characterising various aspects of the project (e.g. resource constraints); setting up project goals; assessing how these goals are supported by the adopted process model, tailoring the process model to suit project goals; using the tailored process model in the project; assessing and fine-tuning the model on an on-going basis.

"The customisation process would be simplified considerably if process models were organised hierarchically, leading from generic models at the top of the hierarchy to specific models at the bottom."(Madhavji 1991) (the CMM does this to an extent).

"[Measurement] helps in making intelligent decisions and improving over time. But measurement must be focused, based upon goals and models" (Basili 1995)

"To improve their software development, organisations need a definition of clear improvement goals, otherwise the improvement activities will turn out to be as chaotic as the development process itself. These improvement goals should support business objectives in the best possible way. For example, it is not recommended to base improvement on a method that prescribes the installation a software configuration management system, while most projects in the organisation fail because of bad requirements management" (Solingen and Berghout 1999). Setting realistic goals means recognising and prioritising which processes need strengthening.

All identified key stakeholders should be involved in the definition of measurement goals. I.e. project team members involved in requirements, their manager and the improvement team members.

Goals should include
- The purpose (what object and why)
- The perspective (what aspect and who)
- The context characteristics.

P12: Establish/implement a process to assess feasibility & external environment relating to project (Sommerville and Sawyer 1997)

The CMM states that assessing the feasibility of a project should include risk assessment, e.g.: "Software risks associated with cost, resource, schedule, and technical aspects of the project are tracked." SPP Activity 13, SPTO, Activity 10.

This process includes the need to define system boundaries as in (Sommerville & Sawyer, 1997). (Curtis *et al.* 1988) found that accurate problem domain knowledge is critical to the success of the projects.
Analysts may need to steer the client away from requirements that cannot be met within the budget and schedule constraints P, Coad and E Yourdon, "Object-Oriented Analysis" in (Thayer and Dorfman 1990).

Patel advocates the use of object oriented technology that can allow both global and local aspects of requirements to be captured i.e. regional (local use cases and commonalities local environments which require analysis) (Patel 1999)

P13: Establish/maintain repeatable requirement traceability process that is project-based

Establishing and maintaining requirements traceability is a central theme in the CMM, yet it is not explicitly modelled. The traceability activities evident in the CMM include the Configuration Management KPA which is specially focussed on tracking requirements. For example, "Software Configuration Management involves identifying the configuration of the software (i.e. selected software work products and their descriptions) as given points in time, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the software life cycle." CMM section 7.6 Software Configuration Management, a key process area for Level 2.

"Inadequate requirements traceability" was cited as a (albeit minor) problem in my process-based requirements research (Hall *et al.* 2002). A strong requirements traceability process may aid other requirements problems cited such as controlling requirements growth and will assist in requirements re-use, however it is important to use the correct traceability method. For example, requirements recycling is supported by methods that separate vertical, horizontal and evolutionary relationships between entities (Knethen *et al.* 2002); If you have a legacy system Sutcliffe states that current methods do not address requirements in a legacy system context. He proposes a model that can cope with the constraints legacy systems place on new requirements and addresses the need to integrate changes resulting from new requirements without introducing errors into acceptable parts of the existing system (Sutcliffe *et al.* 1999).

"Another important concept in the CMM is traceability. Under the CMM all worthwhile software work products are documented, and the documentation design, code and test cases are traced to the source from which they were derived and to the products of the subsequent engineering activity. Requirements traceability provides a means of analysing impact before a change is made, as well as a way to determine what components are affected when processing a change. " Measurements in the CMM include:

Status of each allocated requirement throughout the lifecycle
Change activity of the allocated requirements
Allocated requirements summarized by category."

(Leffingwell and Widrig 2000)

Traceability is understood to mean "a link or definable relationship between entities" (Watkins and Neal 1994), who state that "You can't manage what you can't trace".

The IEEE define traceability as: " (1)The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or mother-subordinate relationship to one another; for example, the degree to which the requirements and design of a given software component match.
(2) The degree to which each element in a software development product establishes its reason for existing; for example, the degree to which each element in a bubble chart references the requirement that it satisfies.
(IEEE std 610.1-1990 in (IEEE 1999)

"[The successful RE team] maintain a requirements traceability matrix to track a requirement from its origin through its specification to its implementation" (Hofmann and Lehner 2001).

P14: Establish a repeatable process to manage complex requirements at project level

Large-scale projects can span many years and different sites can be highly complex. They may need to be highly reliable, safety critical and customized. "One of the pitfalls of systems engineering is to think that a system is simple (i.e. not complex) when we have a very good understanding of its (application) features. An example of such a system is a banking system visualised by the users as a set of automatic teller machines (ATMs). The functions of an ATM are extremely well understood; its applications are trivial transactions. From a system viewpoint, however, we have to worry about a system with a large database of sensitive information with hundreds to thousands of users. With this system come problems related to security and data base concurrency. Virtually all real-time systems are complex because of the constraints on both cycle time and memory resources(Shere 1988).

According to Yourdon, a system is complex if most of the following features apply to the system:

10,000 </= SLOC </= 100,000 (Source lines of Code)
five to twenty programmers over a two to three year period
several subsystems
100 </= number of modules </= 1,000
(Yourdon 1995)

P, Coad and E Yourdon, "Object-Oriented Analysis" in (Thayer and Dorfman 1990). Object Oriented Analysis contains four major principles for managing complexity: abstraction, information hiding, inheritance and methods of organization. Leffingwell recommends that complex systems entail requirements specification for each sub-system, and non-trivial applications, requirements must be captured and recorded in a document database, model or tool (Leffingwell and Widrig 2000)

Techniques such as functional decomposition and input-output analysis reduce complex systems into manageable subsystems but may not help with complex organizational issues (Yu and Mylopoulos 1997). The *i** framework may be helpful in identifying enterprise integration solutions for organisations that have complex technical and human organizational environments.(Yu and Mylopoulos 1997)

P15: Establish a repeatable process to manage vague requirements at project level

The CMM steers companies away from vague requirements with activities such as: "The allocated requirements are reviewed to determine whether they are clearly and properly stated" RM, Activity 1.2

Vague requirements are defined as requirement documentation that is incomplete and flawed. Also called requirements uncertainty (Moynihan 2000) (El Emam and Madhavji 1995). "The whole purpose of the requirements process is to reduce ambiguity in the development process" (Gause and Weinberg 1989).

El Emam & Madhavji talk about 'requirements uncertainty' and define it as "the difference between the amount of knowledge that is required and that is available about the problem and solution domains". "The greater the uncertainty the greater the amount of changes to the requirements engineering documentation (El Emam and Madhavji 1995).

Davis lists 'unambiguous' requirements specified in the software requirements specification (SRS) on the top of his requirements quality list, and states "an SRS is unambiguous if and only if every requirement stated therein has only one possible interpretation (Davis *et al.* 1993). Davis dedicates a section to unambiguous and complete requirements and suggests ways these may be measured and controlled.

P16: Establish a repeatable process to manage requirements growth/change at project level

Concerns functional and non-functional requirements not documented in original specification that result in changes over time, incorporates changeability decay (Arisholm and Sjoberg 2000) "Change is inevitable when computer software is built. And change increases the level of confusion among software engineers who are working on a project. Confusion arises when changes are not analysed before they are made, recorded before they are implemented, reported to those who should be aware that they have occurred, or controlled in a manner that will improve quality and reduce error". Software Engineering, R Pressman, p 66 in (Dorfman and Thayer 1997). "A primary goal of software engineering is to improve the ease with which changes can be accommodated and reduce the amount of effort expended when changes must be made." *Sic*

The CMM covers this extensively, to include:
"Changes to the allocated requirements are reviewed and incorporated into the software project.

1. The impact to existing commitments is assessed, and changes are negotiated as appropriate.

- Changes to commitments made to individuals and groups external to the organization are reviewed with senior management. (Activity 4 Software Project Planning KPA and Activity 3 of Software Project Tracking and Oversight kpa for practices cover commitments made external to the organisation.)
- Changes to commitments within the organization are negotiated with the affected groups. (Software Project Tracking and Oversight KPA for practices covers negotiating changes to commitments.)"

"The CMM recognizes that change is an integral part of software activity in any development project. In place of frozen specifications we instead strive for a stable baseline of requirements that are well elicited, documented and placed into systems that provide support for managing change. Specifically the CMM requires that as understanding of the software improves, changes to the software work products and activities are proposed, analyzed and incorporated as appropriate. Where changes to requirements are needed, they are approved and incorporated before any work products or activities are changed" (Leffingwell and Widrig 2000).

"Requirements continue to be in a state of flux...Many forces affect this ever-changing requirements e.g P, Coad and E Yourdon, "Object-Oriented Analysis" in (Thayer and Dorfman 1990). : customers, competition, regulators, approver, and technology... We have to accept changing requirements as a fact of life, and not condemn them as a product of sloppy thinking" P, Coad and E Yourdon, "Object-Oriented Analysis" in (Thayer and Dorfman 1990). Patel also advocates the use of object oriented technology in his spiral of change model (Patel 1999).

P17: Establish a repeatable process to manage user understanding

Comprehension: People do not know what they want. This does not mean that people do not have a general idea of what the software is for. Rather, they do not begin with a precise and detailed understanding of what functions belong in the software, what the output must be for every possible input, how long each operation should take, how one decision will affect another, an so on.... It is a precise and richly detailed understanding of expected behaviour that is needed to create effective designs and develop correct code. (Faulk, S, "Software Requirements: A Tutorial" in (Dorfman and Thayer 1997).
Laura Scharer, 1981, Pinpointing Requirements in (Thayer and Dorfman 1990) explains that users have different goals and approach to requirements than system analysts. She suggests that although users provide the system definition, the systems people are responsible for it, and that if the user understands their own needs definability is positively affected.
Managing uncertainty in requirements was identified as a major concern to practitioners in El Emam's and Madhavji's field study (1995) – recommendations as to how to help solve this problem include recognising the skill levels required in developers and users and assigning the necessary skills to the project.

P18: Monitor progress of the set requirements goals

Goals are a part of every key process activity in the CMM.
Business goals – having 'set' goals, goals need to be monitored. See P11 'set goals' for further references.
Solingen and Berghout (1999) suggests that goals are reviewed:
The goals should be reviewed and approved by a project team before data collection can actually begin. The review session should focus on:
Do project members agree upon the defined goals, questions and metrics?
Do project members identify any missing or unnecessary definitions?

P19: Agree and document technical and organisational attributes specific to project.

The inclusion of this process is primarily motivated bymy empirical work (see Beecham et al 2003, and Hall et al 2002).

A well defined requirements process leads to a flexible system that is quick to respond to change (e.g. links to resources, traceability, and is cohesive).

"To succeed you must integrate your technical, cognitive, social and organizational processes to suit your project's particular needs and characteristics" (Hofmann and Lehner, 2001)

"One of the most common reasons systems fail is because the definition of system requirements is bad" Laura Scharer, Pinpointing Requirements in (Thayer and Dorfman, 1990)

The process and principles of defining and documenting processes are applied to each of the 5 requirements phases. For example, the documentation phase needs to "define a standard document structure; explain how to use the document, include a summary of the requirements; make a business case for the system; define specialised terms; lay out the document for readability; make document easy to change.

One project management method should be used project wide, e.g. waterfall, spiral, rapid and joint application development, eXtreme Programming (Rule,2001). The CMM also recommends that "A software life cycle with predefined stages of manageable size is identified or defined " in Software Project Planning, Activity 5 (Paulk et al, 1997).

Further references in support of this process are:

(Sommerville and Sawyer, 1997) p.223; (Cugola and Ghezzi, 1998) (Sawyer et al., 1997 4.4); (Pfleeger and Rombach, 1994); (Fayad, 1997) and (Christie, 1999).

A20: Establish a process to review allocated requirements within the project to include software managers and other affected groups

This process is taken direction from the SW-CMM (Paulk *et al.* 1995). It is a CMM activity: RM: Activities Performed, Activity 1: The software engineering group reviews the allocated requirements before they are incorporated into the software project.

1. Incomplete and missing allocated requirements are identified
2. The allocated requirements are reviewed to determine whether they are:
   - Feasible
   - Clearly named properly stated
   - Consistent with each other
   - testable

"Successful teams repeatedly validate and verify requirements with multiple stakeholders. They use peer reviews, scenarios, and walk-throughs to improve the specification throughout the software's life cycle."(Hofmann and Lehner 2001)

"People typically repeat past behaviors, including those that lead to success and those that do not. The organization must ensure that mistakes are not repeated that may have
caused similar initiatives to fail in the past". (McFeeley 1996) section 3.5 "Review Past Improvement Efforts".

According to Davis, a software requirements specification is verifiable if there exist finite, cost effective techniques that can be used to verify that every requirement stated therein is satisfied by the system as built. He states that some requirements are easy to test, whereas others may be difficult to verify – he lists reasons for requirements being difficult and suggests methods for controlling difficult requirements (Davis *et al.* 1993).

"Any engineering process requires feedback and evaluation. Software development is an engineering discipline and measurement is an ideal mechanism for feedback and evaluation.

The measurements and information fed back to developers, managers, customers and the [organisation] help in the understanding and control of the software processes and products and the relationships between them" (Basili 1995).

Appendix M: An example of a requirements process assessment

This example shows how the R-CMM measures the capability of the elicitation phase of requirements. The elicitation phase is just one of the 5 phases represented in the R-CMM. The processes listed in Table 2 define the requirements elicitation phase:

The R-CMM Level 2 Requirements Elicitation Phase

Process Ref.	Process Description
P6	Establish process to **identify skills needs** within *elicitation phase* of the project, e.g. UML, Formal methods
P8	Explore **alternative solutions**, requirements techniques and tools for *the elicitation phase* of project
P10	Establish and maintain process to **involve key stakeholders** in requirements *elicitation phase* of project
P11	**Set realistic goals** to address business requirements and requirements process improvements needs within project
P12	Establish and implement process to **assess feasibility & external environment** of project
P13	Establish and maintain repeatable requirement **traceability process** that is specific to the project
P19	**Agree and document** technical and organisational attributes specific to the *elicitation process* in the project

Table 2: Level 2 R-CMM Elicitation processes

1. Measuring individual processes

The first stage involved in measuring the capability of the requirements process assesses the strength of an individual process. Process P19 in Table 2 is used as an example. This method can be used to assess the strength of any defined process within the R-CMM. Three elements of the process are measured: the approach, the deployment and the application.

Step One. A clear understanding of the process is confirmed

A detailed definition is included with each question. The participant only continues with the assessment if the definition is clearly understood. An example of a process summary is given in Figure 3.

PROCESS 19 "Agree and document technical and organisational attributes specific to the elicitation process of the project".

The requirements elicitation document should show clear links to resources, must be traceable, and must be cohesive.

This document your company produces on how system requirements are discovered should explain how you:

- Consult with stakeholders
- Study existing system documents
- Record requirements rationale
- Gather domain knowledge and document domain constraints
- Define the systems operational environment
- Assess system feasibility
- Agree requirements with stakeholders
- Record any organisational and political considerations and requirements sources
- Use business concerns to guide requirements
- Undertake market studies
- Document technical, cognitive, social and organizational processes that suit your project's particular elicitation needs and characteristics. I.e. explain what techniques and tools are used (e.g. prototype poorly understood requirements, scenarios to elicit requirements, reuse requirements).

include a summary of the requirements; make a business case for the system; define specialised terms; lay out the document for readability; make document easy to change.

A software life cycle with predefined stages of manageable size is identified or defined. One method should be used project wide, e.g. waterfall, spiral, rapid and joint application development, eXtreme Programming (Paulk et al, 1995). A requirements process should also have pre-defined stages.

Figure 3: Process summary for P19

Prior to participating in the questionnaire assessment, participants are told "Please note: you do not have to personally be involved in performing the process – it's enough that you know who performs it to answer the following" (SEI 1996).

Step Two: The Approach to P19 is assessed

The first of the 3 measurement elements is based on the participant's understanding of the company's approach to the process. This encompasses the SW CMM characteristics of demonstrating a commitment to perform and ability to perform the process. Table 3 gives an example of how a participant might respond to the following approach related statements:

APPROACH	Score	
Management Approach	*(Tick one of the options)*	
No management recognition of need		*Poor (0)*
Management has begun to recognise the need		*Weak (2)*
Wide but not complete commitment by management		*Fair (4)*
Some management commitment/some are proactive	✓	*Marginally qualified (6)*
Total management commitment; majority are proactive		*Qualified (8)*
Management provides zealous leadership & commitment		*Outstanding (10)*
Management interest not known		*N/a*
Management interest not believed relevant		*N/a*
Organisational Approach	*(Tick one of the options)*	
No organisational ability/ No organisational commitment		*Poor (0)*
The practice is implemented in one or two projects		*Weak (2)*
Road map for practice implementation defined	✓	*Fair (4)*
Practice implementation under way in parts of the organisation		*Marginally qualified (6)*
Practice established as an integral part of the requirements phase		*Qualified (8)*
Organisational excellence in practice recognised even outside org		*Outstanding (10)*
Organisational approach not known		*N/a*
Organisational approach not believed relevant		*N/a*
Support for Practice	*(Tick one of the options)*	
Practice not evident		*Poor (0)*
Support items for the practice start to be created		*Weak (2)*
Several supporting items for the practice in place		*Fair (4)*
Supporting items in place	✓	*Marginally qualified (6)*
Supporting items encourage and facilitate use of practice		*Qualified (8)*
All support items in place continue to be improved		*Outstanding (10)*
Support for practice not known		*N/a*
Support for practice not believed relevant		*N/a*

Table 3: Generic matrix measuring an organisation's approach to a process

Approach score for process 19: The process *"Agree and document technical and organisational attributes specific to the elicitation phase of the project"* is marginally qualified, i.e. (6 + 4 + 6 / 3 = 5.33)

Step Three: The Deployment of Process 19

This section assesses how a process is deployed in practice. The statements in Table 4 incorporate SW CMM characteristics where each process is analysed, measured and verified. Table 4 shows how a participant might respond to the following statements that relate to how the process is deployed.

DEPLOYMENT	Score	
Use of practice	*(Tick one of the options)*	
No part of the organisation uses the practice		*Poor (0)*
Fragmented or inconsistent use in one or two projects		*Weak (2)*
Less fragmented use; consistency in some projects		*Fair (4)*
Consistent use across most projects	✓	*Marginally qualified (6)*
Deployed in almost all parts of the organisation		*Qualified (8)*
Pervasive/ consistent deployment across all parts of org		*Outstanding (10)*
Use of practice not known		*N/a*
Use of practice not thought relevant		*N/a*
Monitoring of Practice	*(Tick one of the options)*	
No part of the organisation monitors use of practice		*Poor (0)*
Very limited monitoring of use		*Weak (2)*
Monitoring of practice use in some projects		*Fair (4)*
Monitoring of practice use in many projects		*Marginally qualified (6)*
Monitoring of practice use for almost all projects	✓	*Qualified (8)*
Monitoring of practice is continuous across all projects		*Outstanding (10)*
Monitoring of practice not known		*N/a*
Monitoring of practice not thought relevant		*N/a*
Verification of practice	*(Tick one of the options)*	
No part of the organisation verifies use of practice		*Poor (0)*
Very limited verification of deployment		*Weak (2)*
Verification of practice deployment in some projects		*Fair (4)*
Verification of practice deployment in many projects		*Marginally qualified (6)*
Verification of practice deployment in almost all projects	✓	*Qualified (8)*
Verification of practice is continuous across all projects		*Outstanding (10)*
Verification of practice not known		*N/a*
Verification of practice not thought relevant		*N/a*

Table 4: Generic Matrix measuring process deployment

Deployment score for process 19: The responses in this section show that the process *"Agree and document technical and organisational attributes specific to the elicitation phase of the project"* is deployed in a qualified way, i.e. (6 + 8 + 8 / 3 = 7.3).

Step Four: Measuring the application of Process 19

This final dimension measures whether the process goals are appropriate and looks at the effectiveness of the activities performed. These measurements are also characteristics of the SW-CMM.

The statements in Table 5 show how processes are measured to give proof of their value and how they are used throughout the organisation.

RESULTS		Score
Effectiveness of Practice		*(Tick one of the options)*
Ineffective		*Poor (0)*
Some evidence of effectiveness in a few projects		*Weak (2)*
Useful for some projects but not for all		*Fair (4)*
Positive, measurable results over time across many projects	✓	*Marginally qualified (6)*
Positive, measurable results over time across almost all projects		*Qualified (8)*
Requirements exceeded; counsel sought by others		*Outstanding (10)*
Use of practice not known		*N/a*
Rating this practice is not thought relevant		*N/a*
Consistency of Results		*(Tick one of the options)*
Totally random; inconclusive; not measured		*Poor (0)*
Inconsistent results		*Weak (2)*
Consistent and positive results for some projects		*Fair (4)*
Consistently positive results over time across many projects		*Marginally qualified (6)*
Consistently positive results over time across almost all projects	✓	*Qualified (8)*
Requirements exceeded		*Outstanding (10)*
Consistency of results not known		*N/a*
Consistency of results not relevant		*N/a*
Sharing of Results/Best Practice		*(Tick one of the options)*
No practices shared within project,		*Poor (0)*
Some practices shared within project		*Weak (2)*
Most practices shared/applied within project		*Fair (4)*
Practices repeated in many similar projects	✓	*Marginally qualified (6)*
Practices shared throughout all projects		*Qualified (8)*
New practices introduced to support world class results		*Outstanding (10)*
Sharing of this best practice not known		*N/a*
Sharing of this best practice not thought relevant		*N/a*

Table 5: Generic Matrix to establish the strength of process application

Results score for process 19: The responses to this assessment indicate that the results of process ***"Agree and document technical and organisational attributes specific to the elicitation phase of the project"*** is marginally qualified, i.e. ((6 +8 + 6)/3 = 6.6).

Step Five: Combining Process scores to assess the strength of each requirements phase

All three evaluation dimensions and their scoring guidelines are examined simultaneously and all dimensions are equally weighted. Averaging the score of process assessment indicates a level of capability. For example P19 is 'marginally qualified' having received an average score of 6 for its approach, deployment and application, i.e. (5 + 7 + 6 = 18 / 3 = 6).

When all the processes in the requirements phase have been assessed, then a capability for each phase can be obtained. Figure 4 gives an example of a Requirements Phase Assessment sheet. It shows how each measured process is combined to give a score that relates to – in this case – the capability of the elicitation phase of requirements. All the 5 requirements phases are assessed in a similar way.

This assessment gives the following results:

A score for each process
A score for each requirements phase

A score for the requirements process

The validation of the R-CMM highlighted that giving each of the above dimensions the same weighting may not suit some companies. For example, the 'application' section may be considered more important than the 'approach', i.e. if the process proves to be very useful and is being used successfully, management support may not be so important. In this case, a company may decide to place a weighting on the application dimension.

Organisation: ORG_NAME

CMM Level 2 Processes Date:

KRPA: Requirements Elicitation Phase Average Score:.5

(3+ 4 + 5 + 4 + 6 + 7 + 6 = 35 / No of processes (7) = 5)

	List of key processes	1	2	3	4	5	6	7	8	9	10
P6	Establish process to **identify skills needs** within *elicitation phase* of the project, e.g. UML, Formal methods			✓							
P8	Explore **alternative solutions,** requirements techniques and tools for *the elicitation phase* of project				✓						
P10	Establish and maintain process to **involve key stakeholders** in requirements *elicitation phase* of project					✓					
P11	**Set realistic goals** to address business requirements and requirements process improvements needs within the project				✓						
P12	Establish and Implement process to **assess feasibility & external environment** of project						✓				
P13	Establish and maintain repeatable requirement **traceability process** that is specific to the project							✓			
P19	**Agree and document the technical and organisational attributes** specific to the *elicitation process* in the project						✓				

Figure 4: Requirements Phase Assessment sheet.

Section 5 has shown how a process is defined and assessed to establish its strength within the requirements process. In Section 6 I show how this assessment method relates to the SW-CMM.

Appendix N: Expert rating of twenty baseline candidate processes

Frequencies of expert ranking of\ candidate processes at a level 2 (baseline) capability

Process description		Not Needed	Desirable	Essential	Don't know	total
P1:	Follow a written organisational policy for managing the system requirements allocated to the software project	3	7	9	1	20
P2:	Establish project responsibility for analysing the system requirements and allocating them to hardware, software, and other system components	2	2	16	0	20
P3:	Implement training programme to recognise and meet technical and organisational requirements needs within the project	2	10	8	0	20
P4:	Establish process to identify stakeholders within the requirements phase of the project	0	4	16	0	20
P5:	Provide adequate resources and funding for managing the allocated requirements in the project	1	5	14	0	20
P6:	Establish process to identify skills needs within project	2	12	6	0	20
P7:	Institute process to maintain organisational stability within project, e.g. control staff change	3	13	1	3	20
P8:	Explore alternative solutions, requirements techniques and tools for the project	4	10	5	1	20
P9:	Establish/maintain process to reach agreement with customer on requirements for project	0	0	20	0	20
P10:	Establish/maintain process to involve key stakeholders in requirements phase of project	0	2	18	0	20
P11:	Set realistic improvement goals to address problems in the requirements process project	4	14	1	1	20
P12:	Establish/implement process to assess feasibility & external environment of project	1	8	9	2	20
P13:	Establish/maintain repeatable requirement traceability process that is project-based	0	5	14	1	20
P14:	Establish a repeatable process to manage complex requirements at project level	1	6	12	1	20
P15:	Establish a repeatable process to manage vague requirements at project level	2	6	10	2	20
P16:	Establish a repeatable process to manage requirements growth at project level	0	5	14	1	20
P17:	Establish a repeatable process to manage user understanding at project level	0	9	10	1	20
P18:	Monitor progress of the set requirements goals	1	8	10	1	20
P19:	Agree and document technical and organisational attributes specific to project	3	8	9	0	20
P20:	Establish a process to review allocated requirements within the project to include software managers and other affected groups	2	8	10	0	20
	TOTAL	**31**	**142**	**212**	**15**	**400**
	Percentage	**7.75**	**35.5**	**53**	**3.75**	**100**

Appendix O: Personal Publications

Published Journal Papers:

Beecham, S., T. Hall, A. Rainer (2003). "Software Process Improvement Problems in 12 Software Companies: An Empirical Analysis." Empirical Software Engineering 8(1): 7-42.

Hall, T., S. Beecham, A. Rainer (2002). Requirements Problems in Twelve Companies: An Empirical Analysis. IEE Proceedings for Software.

Hall, T., A. Rainer, N. Baddoo, S. Beecham (2001). An Empirical Study of Maintenance Issues within Process Improvement Programmes in Software Industry. IEEE Conf Soft Maint, Florence, Italy.

Conferences

Hall, T., S. Beecham, A. Rainer, (2002). Requirements Problems in Twelve Companies: An Empirical Analysis. EASE, Keele.

Britton, C., M. Kutar, Anthony, S, Barker, T, Beecham, S, Wilkinson, V (2002). An Empirical Study of User Preference and Performance with UML Diagrams. IEEE CS International Symposium on Human-Centric Computing Languages and Environments, Arlington, VA, USA., IEEE Computer Society.

Publications (in review)

Beecham, S., T. Hall, A. Rainer (2003). "Building a Requirements Process Improvement Model." Software Process: Improvement and Practice. In review.

Beecham, S., T. Hall, A. Rainer (2003). "Defining a Requirements Process Improvement Model." Software Quality Journal. In review.

Beecham, S., T. Hall, C. Britton, M. Cottee, A. Rainer (2003). "Using an expert panel to validate a requirements software process improvement model." IEEE Transactions on Software Engineering. In review.

Technical Reports

Beecham, S. (2002). Correspondence Analysis and inertia explained - A worked example. TR No. 365. Dept of Computer Science. Hatfield, University of Hertfordshire.

Beecham, S. (2002). Cohen's Kappa Measure of Agreement - A worked example. TR No. 366. Dept of Computer Science. Hatfield, University of Hertfordshire.

Beecham, S., T. Hall, A. Rainer (2003). Building a Requirements Process Improvement Model. TR No. 378. Dept of Computer Science. Hatfield, University of Hertfordshire.

Beecham, S., T. Hall, A. Rainer (2003). Defining a Requirements Process Improvement Model. TR 379. Dept of Computer Science. Hatfield, University of Hertfordshire.

Beecham, S., T. Hall, A. Rainer (2003). Assessing the strength of a requirements process. TR No. 381. Dept of Computer Science. Hatfield, University of Hertfordshire.

Beecham, S., T. Hall, C. Britton, M. Cottee, A. Rainer (2003). Validating a Requirements Process Improvement Model, TR No. 373. Dept of Computer Science. University of Hertfordshire.

Appendix P: References used in Appendices

ami. 1992. A Quantitative Approach to Software Management. South Bank University.

Arisholm, E. and Sjoberg, D. I. K. 2000. Towards a framework for empirical assessment of changeability decay. Journal of Systems and Software (53): 3-14.

Baddoo, N. and Hall, T. 2002. Motivators of Software Process Improvement: an analysis of practitioners' views. The Journal of Systems and Software 62 (2): 85-96.

Basili, V. R. 1995. Measurement Frameworks. Ed(s): N. Fenton, R. Whitty and Y. Iizuka. Software Quality Assurance and Measurement: A Worldwide Perspective International Thomson Computer Press, London.

Basili, V. R. and Rombach, H. D. 1988. The Tame Project: Towards Improvement-oriented software environments. IEEE Transactions on Software Engineering 14 (6): 758-773.

Boehm, B. (2001). Using Win-Win for Requirements Negotiation: a mini tutorial. London,

Boehm, B. W. 1981. Software Engineering Economics. Englewood Cliffs, Prentice-Hall, Inc.

Britton, C. 2000. Object-oriented system development : a gentle introduction. New York; London, McGraw-Hill.

Christie, A. M. 1999. Simulation in support of CMM-based process improvement. Journal of Systems and Software 46 (2-3): 107-112.

Coad, P. and Yourdon, E. 1990. Object Oriented Analysis. Ed(s): R. H. Thayer and M. Dorfman. System and Software Requirements Engineering IEEE Computer Society Press, Los Alamitos, CA.

Cohn, M. and Ford, D. 2003. Introducing an agile process to an organisation. IEEE Computer: 74-78.

Cottengim, D. 2002. Prerequisites for Success: Why Process Improvement Programs Fail. CrossTalk The Journal of Defense Software Engineering: 26-30.

Cresswell, J. 1998. Quality Inquiry and Research Design: Choosing among five traditions. Thousand Oaks, Sage Publications.

Cugola, G. and Ghezzi, C. 1998. Software Processes: a Restrospective and a Path to the Future. Software Process - Improvement and Practice 4 (3): 101-123.

Curtis, B., Hefley, W. E., Miller, S. and Konrad, M. 1995. People Capability Maturity Model. CMU/SEI-95-MM-02. Software Engineering Institute, Carnegie Mellon University.

Curtis, B., Krasner, H. and Iscoe, N. 1988. A Field Study of the Software Design Process for Large systems. Communications of the ACM 31 (11): pp 1268-1287.

Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A. and Theofanos, M. 1993. Identifying and Measuring Quality in a Software Requirements Specification. In Proceedings of the First International Software Metrics Symposium: 141-152.

Dooley, K., Subra, A. and Anderson, J. 2001. Maturity and its impact on new product development project performance. Research in Engineering Design 13 (1): 23-29.

Dorfman, M. and Thayer, R. H. 1997. Software Engineering. Los Alamitos, CA, IEEE Computer Society Press.

El Emam, K. and Madhavji, N. H. 1995. A Field Study of Requirements Engineering Practices in Information Systems Development. Second IEEE International Symposium on Requirements Engineering: 68-80.

El Emam, K., Quintin, S. and Madhavji, N. H. 1996. User Participation in the Requirements Engineering Process: an Empirical Study. Requirements Engineering Journal 1 (1): 4-26.

Fayad, M. E. 1997. Software Development Process: A Necessary Evil. Communications of the ACM 40 (9): 101-103.

Gause, D. C. and Weinberg, G. M. 1989. Exploring requirements: quality before design. New York, NY, Dorset House Publishing.

Gilb, T. 1996. Level 6: why we can't get there from here. IEEE Software 13 (1): 97-98, 103.

Gross, D. and Yu, E. 2001. From Non-Functional Requirements to Design through Patterns. Requirements Engineering (6): 18-36.

Hall, T., Beecham, S. and Rainer, A. 2002. Requirements Problems in Twelve Companies: An Empirical Analysis. IEE Proceedings for Software 149: 153-160.

Hofmann, H. F. and Lehner, F. 2001. Requirements Engineering as a Success Factor in Software Projects. IEEE Software: 58-66.

Hughes, J. O. B., J.; Rodden, T.; Rouncefield, M.; Sommerville, I. and 27-34, P. s. 1995. Presenting ethnography in the requirements process. Proceedings of

1995 IEEE International Symposium on Requirements Engineering (RE'9: 27-34.

Humphrey, W. S. 2002. Three Process Perspectives: Organizations, Teams, and People. Annuls of Software Engineering 14: 39-72.

IEEE. 1998. IEEE Guide to Software Requirements Specification. IEEE Press.

IEEE. 1999. IEEE Standards Software Engineering, 1999 Edition. Volume One: Customer and Terminology Standards. The Institute of Electrical and Electronics Engineers, Inc.

Jackson, M. 1995. Problems and Requirements. Second IEEE International Symposium on Requirements Engineering: 2-8.

Jacobs, S. 1999. Introducing Measurable Quality Requirements: A Case Study. 4th IEEE International Symposium on Requirements Engineering.

Jalote, P. 1997. An Integrated Approach to Software Engineering: Second Edition. New York.

Kitchenham, B. 1995. Using function points for software cost estimation. Ed(s): N. Fenton, R. Whitty and Y. Iizuka. Software Quality Assurance and Measurement: A Worldwide Perspective International Thomson Computer Press, London.

Knethen, A. v., Paech, B., Kiedaisch, F. and Houdek, F. 2002. Systematic Requirements Recycling through Abstraction and Traceability. Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02): 273-281.

Kotonya, G. and Sommerville, I. 1998. Requirements Engineering : processes and techniques. Chichester, John Wiley & Sons.

Lauesen, S. and Vinter, O. 2001. Preventing Requirement Defects: An Experiment in Process Improvement. Requirements Engineering Journal 6 (1): 37-50.

Leffingwell, D. and Widrig, D. 2000. Managing software requirements: a unified approach. Reading, Massachusetts, Addison-Wesley.

Linberg, K. R. 1999. Software developer perceptions about software project failure: a case study. Journal of Systems and Software 49: 177-192.

Lindland, O. V., Sindre, G. and Solvberg, A. 1994. Understanding Quality in Conceptual Modeling. IEEE Software 11 (2): 42-49.

Loucopoulos, P. and Karakostas, V. 1995. System requirements engineering. London, UK, McGraw-Hill Book Company Europe.

Madhavji, N. H. 1991. The process cycle. Software Engineering Journal 6 (5): 234-242.

Maiden, N. and Gizikis, A. 2001. Where do Requirements Come From? IEEE Software (September/October): 10-12.

Marcoulides, G. and Hershberger, S. 1997. Multivariate statistical methods : a first course. Mahwah, New Jersey, Lawrence Erlbaum Associates, Publishers.

McFeeley, B. 1996. IDEAL: A User's Guide for Software Process Improvement. CMU/SEI-96-HB-001. Software Engineering Instutute, Carnegie Mellon University.

Middleton, P. and McCollum, B. 2001. Management of process improvement by prescription. Journal of Systems and Software (57): 9-19.

Morisio, M. and Torchiano, M. 2002. Definition and Classification of COTS: A Proposal. Ed(s). COTS-Based Software Systems : First International Conference, ICCBSS 2002, Orlando, FL, USA, February 4-6, Proceedings Springer-Verlag Heidelberg.Page: 165.

Moynihan, T. 2000. Coping with 'requirements-uncertainty': the theories-of-action of experienced IS/software project managers. Journal of Systems and Software (53): 99-109.

Onvlee, J. 1995. Use of function points for estimation and contracts. Ed(s): N. Fenton, R. Whitty and Y. Iizuka. Software Quality Assurance and Measurement: A Worldwide Perspective International Thomson Computer Press, London.

Patel, N. V. 1999. The Spiral of Change Model for Coping with Changing and Ongoing Requirements. Requirements Engineering 4 (4): 77-84.

Paulk, M. C. 1994. A Comparison of ISO 9001 and the Capability Maturity Model for Software. CMU/SEI-94-TR-12. Carnegie Mellon University: 78.

Paulk, M. C., Weber, C. V., Curtis, B. and Chrissis, M. B. 1995. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, Massachusetts, Addison Wesley Longman, Inc.

Pearsall, J.1999. Concise Oxford Dictionary. Tenth Edition, Revised. New York, Oxford University Press.

Pfleeger, S. L. and Rombach, H. D. 1994. Measurement based Process Improvement. IEEE Software 11 (4): 9-11.

Rule, G. P. 2001. Using measures to understand requirements. European Software Conference on Metrics (ESCOM-2001) Conference Proceedings.

Sawyer, P., Sommerville, I. and Viller, S. 1997. Requirements Process Improvement Through the Phased Introduction of Good Practice. Software Process Improvement and Practice 3 (1).

Scharer, L. 1990. Pinpointing Requirements. Ed(s): R. H. Thayer and M. Dorfman. System and Software Requirements Engineering: IEEE Computer Society Press Tutorial IEEE Computer Society Press, Los Alamitos, CA.

SEI. 1996. A Description of the Systems Engineering Capability Maturity Model Appraisal Method Version 1.1. Report. SECMM-94-06CMU/SEI-96-HB-004 v1.1. Software Engineering Institute, Carnegie Mellon University.

SEI. 2002. Capability Maturity Model ® Integration (CMMI SM), Version 1.1. CMMI SM for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1). Continuous Representation. Technical Report. CMU/SEI-2002-TR-01. ESC-TR-2002-011. Carnegie-Mellon Univ: 1-724.

Shere, K. D. 1988. Software Engineering and Management. Englewood Cliffs, N.J., Prentice Hall, Inc.

Solingen, R. v. and Berghout, E. 1999. The Goal/Question/Metric Method: a practical guide for quality improvement of software development. Maidenhead, UK, McGraw-Hill Publishing Company.

Sommerville, I. 2001. Software Engineering. Wokingham, Addison-Wesley Publishing Company.

Sommerville, I. and Sawyer, P. 1997. Requirements Engineering A good practice guide. Chichester, John Wiley & Sons Ltd.

StandishGroup. 1995. Chaos Report.

Stricker, C. 1995. Evaluating effort prediction systems. Ed(s): N. Fenton, R. Whitty and Y. Iizuka. Software Quality Assurance and Measurement: A Worldwide Perspective International Thomson Computer Press, London.

Sutcliffe, A. G., Economou, A. and Markis, P. 1999. Tracing Requirements Errors to Problems in the Requirements Engineering Process. Requirements Engineering (4): 132-151.

Thayer, R. H. and Dorfman, M. 1990. System and Software Requirements Engineering. Los Alamitos, CA, IEEE Computer Society Press.

Tingey, M. O. 1997. Comparing ISO 9000, Malcolm Baldrige and the SEI CMM for Software. A reference and section guide. Upper Saddle River, Prentice Hall PTR.

Watkins, R. and Neal, M. 1994. Why and How of Requirements Tracing. IEEE Software (July): 104-106.

Weinberg, G. M. 1998. The psychology of computer programming. New York, NY, Dorset House Publishing.

Weller, S. C. and Romney, A. K. 1990. Metric Scaling: Correspondence Analysis. Newbury Park, California, Sage Publications Inc.

Yourdon, E. 1995. Techniques of Program Structure and Design. Englewood Cliffs, N.J., Prentice-Hall, Inc.

Yu, E. S. K. and Mylopoulos, J. 1997. Modelling Organizational Issues for Enterprise Integration. International Conference on Enterprise Integration and Modelling Technology.