Building a Requirements Process Improvement Model

Sarah Beecham, Tracy Hall and Austen Rainer

Technical Report No 378

Department of Computer Science

Faculty of Engineering and Information Sciences

February 2003

This technical report gives a step-by-step guide to developing a requirements process improvement model. The model is goal driven, and practitioners are directed towards key processes that define and prioritise requirements activities. The requirements model is a specialisation of the established Software Engineering Institute's (SEI) Capability Maturity Model (SW CMM). Best practices from the requirements literature as well as the SW CMM are used to inform the requirements process improvement model. All processes listed address requirements problems raised by practitioners in our empirical study (Hall et al., 2002b; Beecham et al., 2003b).

Detailing the approach to model development will help the research community and practitioners in their own Software Process Improvement (SPI) efforts. Understanding more about the underlying structure will enable the reader to learn from previous experiences, replicate and adapt the model to meet their own improvement goals. Much of the research on process improvement provides guidelines and new models but omit the many steps involved in their generation. Learning more about how the model evolved should help practitioners to interpret key practices, take more control over their processes and create their own tailored improvement models.

1. Introduction

This report is primarily concerned with the process of constructing a model that is based on a recognised framework. "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). By outlining the activities involved in building the model we aim to understand the model's underlying characteristics and gain an insight into how it might successfully be employed in practice. Also showing how the model evolves allows us to explore how closely we keep to our objective of retaining the structure of an existing framework. There is very little in the literature to guide this work as "model development is an area in need of further research" (Eriksson, 2003).
The importance of this work is outlined by Pidd who states "students need to realise that learning the skills of modeling may be more important than learning about models and in his paper he pleas "for some serious research about how people go about their modeling".[Pidd, 1999 #189].

Pidd (1999), whose work is in the field of Operations Research and Management Science, suggests that models have two main uses. (1) ‘they are used to help people explore the possible consequences of actions before they take them; and (2) ‘they are used for routine decision support where the models form an essential and automatic part of the management and control of an organisation.’

Although Pidd states that the model has two distinct uses, our model is a hybrid of the two. It is primarily a tool for assessing the current status of the requirements process and is an aid to thinking about how best to apply recommended practices (as in 1). Yet the model supports decision making by regulating the order in which processes are implemented, as dictated by the maturity structure (as in 2).

The model we are building is "an external and explicit representation of part of reality as seen by the people who wish to use that model to understand, to change, to manage, and to control that part of reality in some way or other". This requirements process improvement model is external and explicit allowing others to examine the perceived process, challenge it, change it (if necessary) and agree to a course of action.

2. The Requirements CMM takes its characteristics from the SW CMM.

In line with the SW CMM (Paulk *et al.*, 1995), the Requirements Capability Maturity Model (Requirements CMM[1]) defines processes at incremental levels of maturity. Maturity levels are characterised by sets of requirements processes that are key to software development. Development of a specialised requirements CMM is primarily motivated by our empirical research into Software Process Improvement (SPI) activities in the UK (Hall *et al.*, 2002b; Beecham *et al.*, 2003b). Our study highlights requirements as a major problem for practitioners despite all companies using the SW CMM to assist them in their software process improvement efforts. Close examination of the Software CMM reveals that although the model does not include all key requirements processes, many of the requirements problems raised by practitioners in our study are addressed. One of the aims of the Requirements CMM, therefore, is to highlight the requirements practices that appear buried in the all-encompassing Software CMM.

Many companies throughout the world use the SW CMM as their software process improvement model. Results of using this method are generally positive with improved processes leading to higher quality software. Therefore we heed the words of Humphrey (a main proponent of the (SW CMM)),

"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".

---

[1] ®CMM is registered in the U.S. Patent and Trademark Office. Accuracy and interpretation of this document are the responsibility of the University of Hertfordshire, Centre for Empirical Software Process Research. Carnegie Mellon University has not participated in this publication.

By harnessing solutions that are in the public domain we aim to build on other people's work.

In agreement with the SW CMM, for example, a company with an immature requirements process is likely to have very few standards in place and could be viewed as having an ad-hoc requirements process. On the other hand, a company with a mature requirements process will follow a set standard that produces a predictable and stable output. Having a reliable requirements 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 requirements process can cope with changes and enhancements with minimal disruption. The Requirements CMM is designed to work with the SW CMM improvement programme to evaluate, understand and identify potential weaknesses in the existing requirements process. To evaluate these strengths and weaknesses the model includes a method for 'assessing' requirements process maturity levels.

3. A generic model

The purpose and the structure of the Requirements CMM is explained in this report.

The content and behaviour of the model is covered in (Beecham *et al.*, 2003a)

3.1 The purpose

In agreement with [Pidd, 1999 #189], we believe the model is no substitute for thought and deliberation and as such should have the following characteristics:

- The model is external and explicit (allowing for external examination)
- The model is a 'representation' of the real world – it should not be as complete or complex as the world they represent - it is a surrogate form of reality that can safely be explored and manipulated
- The representaiton is partial and is governed by our intended use. It should be complete enough to understand, to change to manage and control the key requirements processes at different levels of maturity.
- The modelling is goal-oriented.

The goals of the Requirements CMM are to enhance and augment the Software CMM by guiding practitioners to:

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

3.2. The structure

Stages in Model Development

In order to create a model that is replicable and verifiable we apply a formal approach to development. The first stage in model development therefore is to set the criteria for success using a rule-based framework (Rossi, 1999; Koubarakis and Plexousakis, 2002). These rules guide model development by providing goals to work towards. Also, by clearly defining our objectives and model boundaries we create a framework against which we can later evaluate the model. This addresses a problem highlighted by Kitchenham et al, that "there is an urgent need to address the problem of model evaluation" (Kitchenham *et al.*, 2002).

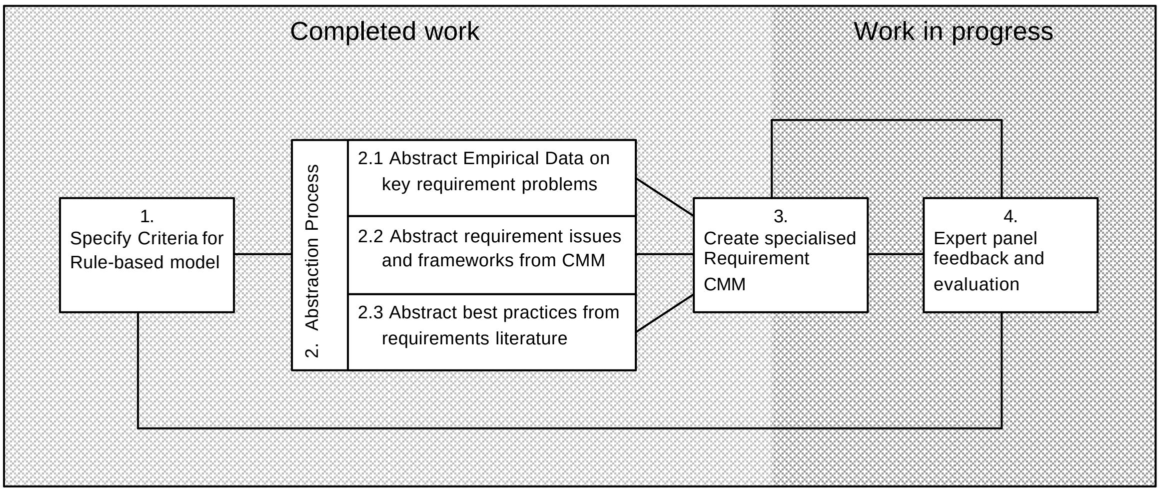

Figure 1: Initial activities involved in building a specialised Requirements CMM

Figure 1 shows the stages involved in creating a rule-based model. First we specify the criteria for model (see Table 1). In stage 2 we abstract the data that will populate the model using three sources. Then we produce a specialised requirements process model. This third stage considers how best to present the data (e.g in a way that is clear, appropriate for the users and easy to use). We are currently in the final stage (stage 4) of model development having sent out a prototype model along with a questionnaire to a group of experts.

Step 1: Specify criteria for rule-based model

Model development is initiated by creating and agreeing the criteria (Rossi 1999; Koubarakis and Plaxousakis, 2002). This formalises the model and sets out rules to create a firm foundation and provides a structure for the building process. The criteria for success given in Appendix 1 initiated the model and created a working framework. This rule-based development technique is particularly relevant to modelling processes (Madhavji, 1991). Fifteen areas are considered when building the model:

Table 1: Criteria for the Requirements CMM Development

Adherence to CMM	Assumptions	Boundaries	Clarity	Consistency
Detail	Ease of Use	Extendable	Focus	Glossary
Granularity	Tailorable	Understandable	Uniqueness	Verifiable

See Appendix 1 for a list of success criteria including related rules and purpose.

Step 2: Abstract key requirement issues

Emulating proven modelling strategies. e.g. (Abdel-Hamid and Madnick, 1989; Dybå, 2000), we used our empirical data together with the requirements literature to inform the requirements CMM.

Step 2.1 Abstract key requirement issues from the empirical data

Model development was driven by the requirements issues raised by practitioners in our empirical study of 12 software development companies (Hall *et al.*, 2002a). We categorised the requirements problems into organisational and technical requirements problems (see Table 3 and 4 in Appendix 2 for breakdown of problems).

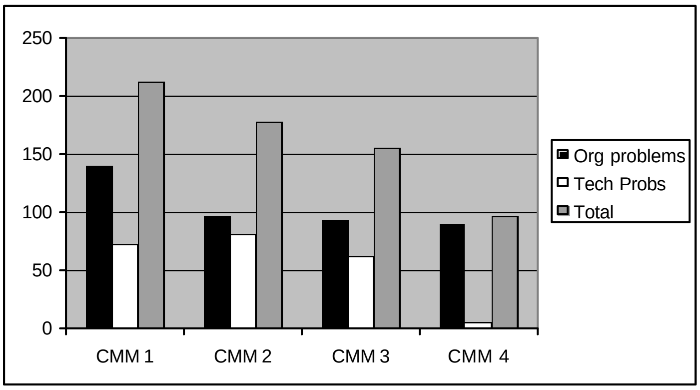

Figure 2: Requirements problems by CMM Maturity Level

The bar chart in Figure 2 separates the organisational and technical requirement problems as recorded in 45 focus group transcripts. The requirements problems are classified by the CMM level of the company reporting the problems. A chi-squared result of $X^2 = 9.38$, df = 3, p = 0.02 indicates that there is a significant association between CMM maturity and diminishing technical problems. However, organisational requirements problems (the black bar) appear untouched by the improvement program (data is taken from table 2 in Appendix 2).

Figure 2 indicates that as companies mature they have fewer problems with their requirements process in general. Yet the organisational problems within the requirements process do not diminish with overall capability. Viewing the requirements process as a whole (the total figure) masks this problem. Therefore, the Requirements CMM will describe the 'organisational' requirements problems separately to the 'technical' requirements problems to ensure the more difficult 'organisational' processes are not overloooked. For a more detailed analysis of the problems raised in our empirical study please refer to (Hall *et al.*, 2002b) and (Beecham *et al.*, 2003b).

Step 2.2: Abstract requirement guidelines and model frameworks from the CMM

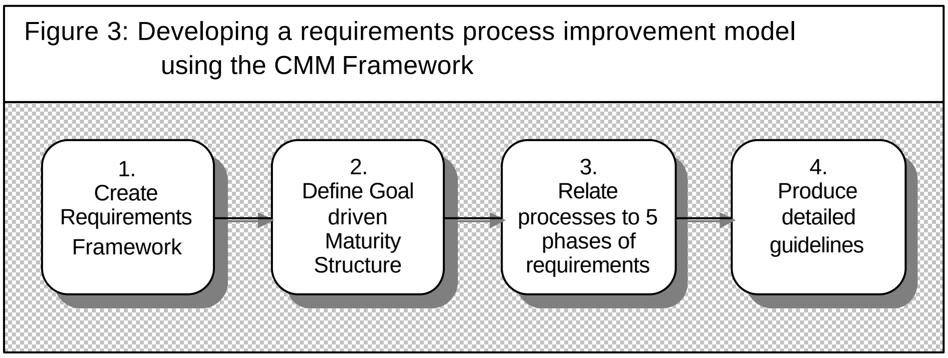

Figure 3: Requirement guidelines and model frameworks abstracted from the Software CMM

Figure 3 shows the 4 stages involved in abstracting requirements related issues from the Software CMM.

Step 2.2.1: The Requirements Framework (CMM Abstraction Stage 1)

A high level view is taken initially where the 5 levels of requirements process maturity grow out of the SW CMM (Figure 4). Figure 4 indicates how the Requirements CMM retains the integrity of the SW CMM and adheres to our objectives. The 5 level structure is retained to reflect different capability goals.

The Requirements CMM should retain the 5 levels of maturity as set out in the SW CMM and Paulk et al (1995) point out the benefit of this structure:

> "The purpose of the CMM is to describe good management and engineering practices as structured by the maturity framework. "

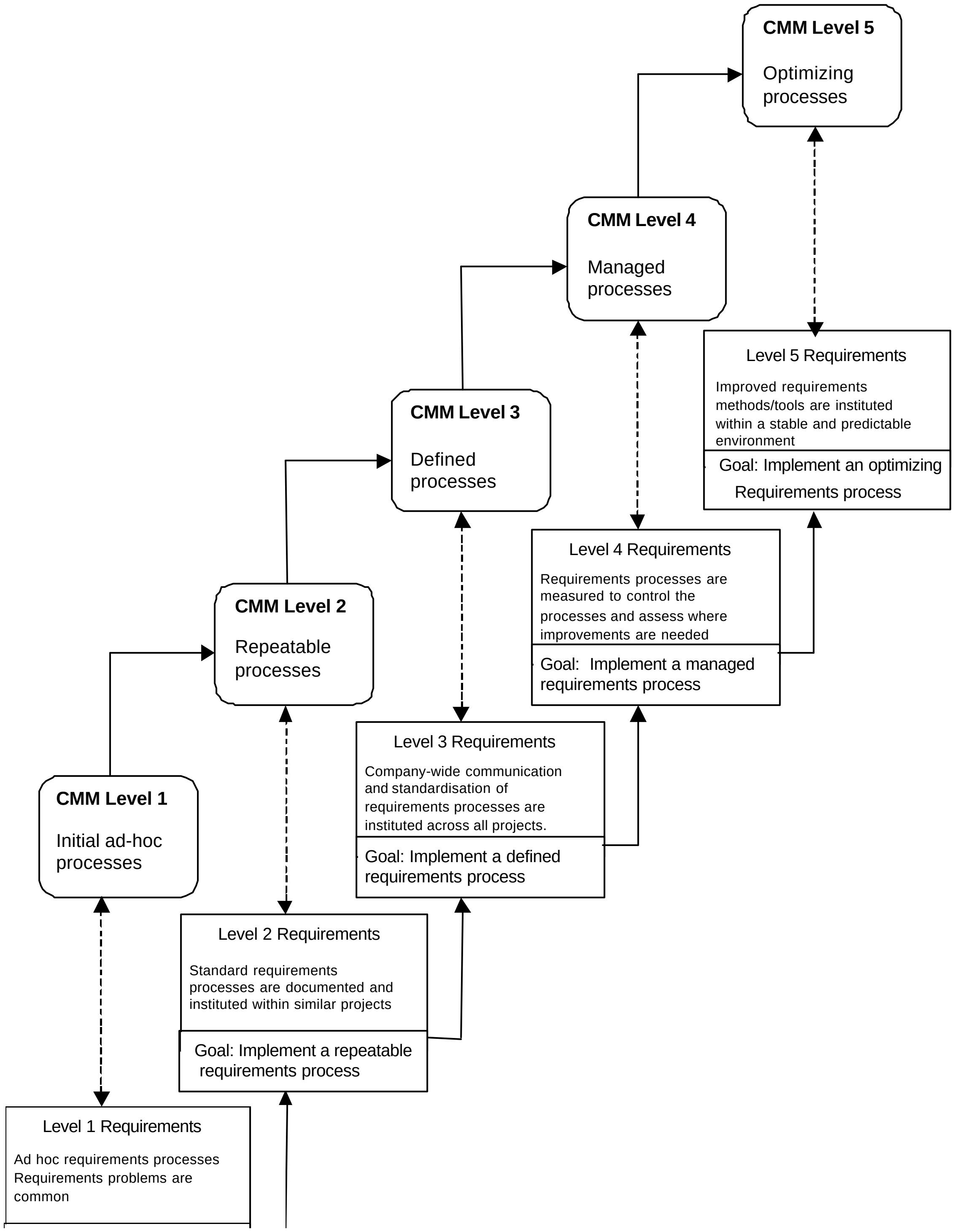

Figure 4: The Requirements CMM Framework

Step 2.2.2: Define Goal driven Maturity Structure (CMM Abstraction Stage 2)

To ensure consistency is maintained within existing maturity levels, it is necessary to explore and understand the underlying structure. Figure 5 is adapted from Paulk (1997) where constituent parts are decomposed and requirements features are added. For example, figure 5 shows how requirements maturity levels indicate process capability and how CMM concepts and empirical findings underpin these maturity levels. Each maturity level (with the exception of level 1) is made up of key requirements processes. When in place, these processes address clearly defined goals. The model is generic to allow for individual company needs.

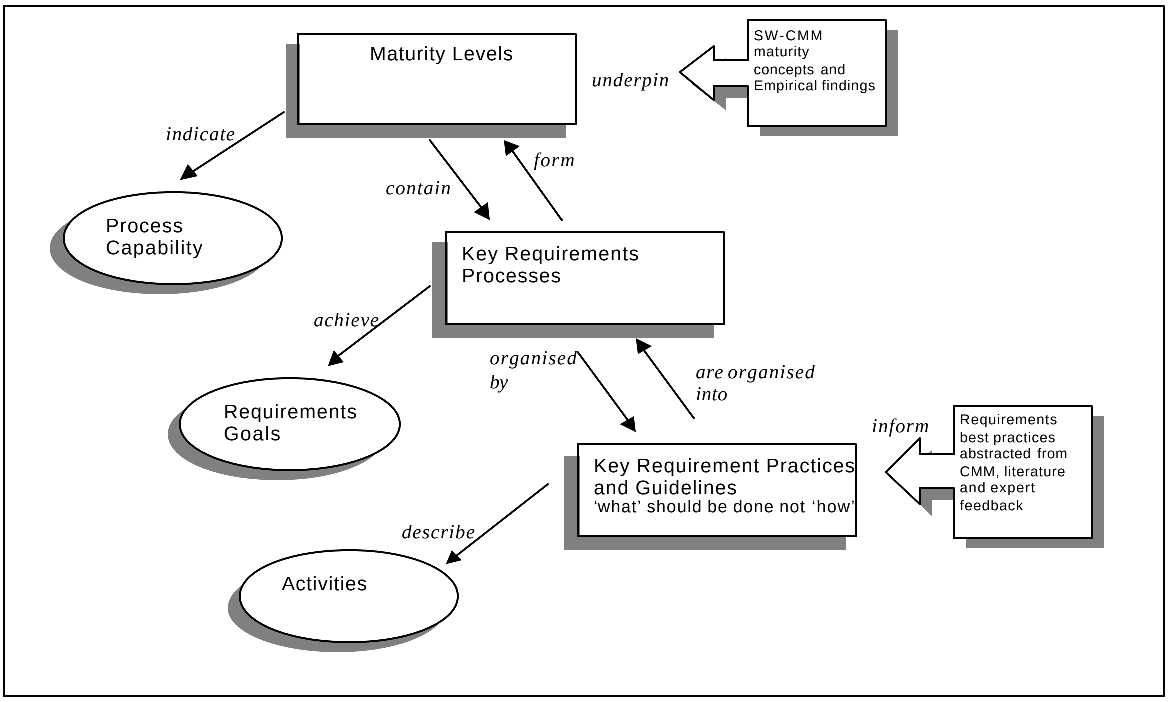

Figure 5: The Requirements CMM Structure (Adapted from Paulk, 1997: p.31)

Step 2.2.3: Relate processes to 5 phases of requirements (CMM Abstraction Stage 3)

Understanding the underlying structure helps to decompose the requirements framework (Figures 4 & 5) into individual maturity levels. Figure 6 in Appendix 3 shows how the Software CMM maturity concept is retained and introduces a Level 2 Requirements CMM that is goal driven. The goals are determined by the CMM level characteristics, i.e. the Level 2 goal is to implement a 'repeatable' requirements process. Appendix 4 gives a brief overview of the Goal Question Metric approach to process improvement.

Step 2.2.4: Produce detailed Guidelines (CMM Abstraction Stage 4)

A criticism of the software CMM is that it does not provide sufficient examples and specific guidelines to help companies with their process improvement (e.g.(Lauesen and Vinter, 2001; Potter and Sakry, 2001).). We have addressed this problem by taking each key requirements process (as listed in Figure 6 in Appendix 3) and extending the process into a detailed guideline. The guideline model retains the Goal Question Metric approach to process improvement, as the process in the higher level model (Figure 6) becomes the 'goal' in the finer grained model example given in Figure 7. 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 general 'descriptive' model that has a more universal application.

Step 2.3: Abstracting best practice from the literature

We have used seminal papers such as (Davis *et al.*, 1993) and (Krogstie, 1998) to define technical requirements processes and have looked to works on qualitative aspects of software improvement to add an organisational perspective to our model (e.g. (Perry *et al.*, 1994). Previous work on re-architecting the CMM and general model design practices have been used to guide development e.g. (Abdel-Hamid and Madnick, 1989; Madhavji, 1991; Christie, 1999; Dybå, 2000; *Ferraiolo*, 2002; Koubarakis and Plexousakis, 2002). We have abstracted lessons learned from using the CMM in practice from both the research community (in the form of case and field studies) and practitioners (in the form of experience reports) e.g. (El Emam *et al.*, 1996; Smith, 1998; Hofmann and Lehner, 2001). These studies help to highlight where current strengths and weaknesses are in CMM requirements' support. The idea of creating a requirements process improvement model based on the CMM is not new, and we have found the parallel work of Sommerville and Sawyer particularly helpful (Sawyer *et al.*, 1997; Sommerville and Sawyer, 1997).

CMM adaptation has also been recorded in areas outside of Software Development such as business strategic planning, e.g. (Hackos, 1997). Here Hackos utilises the CMM framework to establish goals and measurable objectives in the business environment. It has been used as a business model as it "contains the fundamental organizational and management elements for any organisation"(*Ferraiolo*, 2002). Another application of the CMM is found in the "IT Service Capability Maturity Model" that uses the maturity characteristics to guide companies towards providing an optimised service. (Neissink *et al.*, 2002). We note here the cycle of improvement methodologies. Early development of software process improvement methods were influenced by the practices in the manufacturing industry where the importance of statistical process controls were recognised post World War II (Deming, 1982) in (Humphrey, 1989). The fact that the business world is now recognising the strengths of software engineering improvement methods such as the CMM suggests that the software industry itself is maturing.

Step 3: Create a specialised Requirements CMM

The culmination of the previous strands of work described in section 1 and 2 is realised in a set of frameworks and guidelines that focus on requirements process improvement. We have taken the CMM level characteristics and applied them to a specialised requirements CMM. The requirements CMM is a composite model that takes on the characteristics of the SW CMM and uses a Goal Question Metric (GQM) approach to guide the user towards identifying and realising set goals.

Identifying goals prior to following guidelines helps process improvement. The GQM method formalises this goal-focussed view of requirements and is "aimed at providing a basis for corporate learning and improvement" (Basili and Romach, 1988). The GQM paradigm helps to focus on the requirements process and is intuitive having gained widespread acceptance as a way to measure processes in the software engineering community, e.g.(Pfleeger and Rombach, 1994; De Panfilis *et al.*, 1997; Mashiko and Basili, 1997; Gresse and Briand, 1998). It has even been used specifically in the field of measuring the requirements management key process area (KPA) in the SW CMM (Loconsole, 2001).

The Requirements CMM uses the GQM paradigm in the following way:

1. Goals: Each maturity level has a specific goal to work towards. The Software CMM sets the maturity goals for each level of requirements process maturity, e.g. the Level 2 Goal is to 'implement a "repeatable" requirements process). Goals become more defined in the guideline model where processes from the higher level model become goals in the finer grained guideline model (see figures 6 and 7 in Appendix 3 for examples of a goal and sub-goal).

2. Questions to answer within a recognised requirements framework. 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 (Dorfman and Thayer, 1997; Pressman, 2001). Each of the maturity stages will view requirements in these phases to help with implementation.

3. Requirements Processes that reflect maturity level characteristics. For example, we have identified 20 requirements processes that we believe are key to establishing a Level 2 capability. Each process relates to at least one requirements phase as defined in the 'questions'. Each level of maturity (i.e. Levels 2, 3, 4 and 5) will have their own set of recommended processes.

4. Metrics to assess the strengths and weaknesses of each process. We adopt a method used by a high level maturity company to evaluate how well implementation of each process is perceived to be. Each process is measured in turn to assess how well it has been implemented in practice. This form of assessment can be extended to assess the strength of sub-processes (e.g. in guidelines given in Appendix 3, Figure 7) should a finer grained analysis be required. Details of how to apply this form of assessment are included in another Technical Report that is not yet published.

The GQM method can be used to represent processes as elements of the lifecycle model (i.e. it does not have to model the whole of software development). Improving requirements processes is therefore ideally suited to this form of modelling. For clarity, the Requirements CMM looks initially at the processes involved (i.e. the 'Goal/Question → Processes' as modelled in Figure 6 and 7) before applying any form of measurement. The final representation of the Requirements CMM is viewed within a 'Goal/Question/Process → Metric' framework.

Step 4: Validate the model

This stage constitutes work in progress. We emulate previous studies such as (Dybå, 2000) and contacted a group of experts who agreed to validate our model. The experts represent four areas of expertise:

	Software Process Improvement	Requirements
Practitioners	✓	✓
Academics	✓	✓

A questionnaire was designed to query all aspects of the model listed in our criteria for success (as listed in Appendix 1). Responses from this questionnaire will help to validate the model and may result in making some changes to the current presentation and content. We accept that "building models is relatively easy – empirically validating them in the real world is hard" (Christie, 1999).

3: Conclusion

We have shown the many stages involved in developing a specialised requirements process improvement model from a Software CMM framework. We took one level of the Software CMM (Level 2) and broke down the requirements process into 20 key processes. The Requirements CMM breaks away from a static/linear view of the requirements lifecycle to create a model that is goal and problem driven.

The CMM framework is therefore transformed into a simplified model that relates goals and problems to individual requirements practices. The model directs practitioners to examine their requirements process in a systematic and detailed way. The Requirements CMM includes some Software CMM best practices together with additional requirements processes that are outside the scope of the Software CMM. The result is a cohesive and comprehensive model that reflects requirements best practices at incremental levels of capability.

In disagreement with some criticisms levelled at the Software CMM, we believe that the model covers most of the requirement process activities required to create a mature capability. However, in its current form the Software CMM addresses all key software development activities that, although important, do not apply to requirements. It is therefore difficult to isolate and identify all the activities required to meet precise requirement goals.

It is intended that the actual processes involved in developing the model, as outlined in this report, will provide a foundation for future development in the area of requirements process improvement.

Acknowledgements: We are sincerely grateful to all the companies and practitioners their participation in the ‘Managing Practitioner impact on Processes and Products’ (PPP) project. The PPP project is funded by the UK’s Engineering and Physical Science Research Council, under grant number EPSRC GRL91962. We are also grateful to Pete Sawyer for his support in the project.

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.

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. and Romach, H. D. 1988. The Tame Project: Towards Improvement-oriented software environments. IEEE Transactions on Software Engineering, 14 (6): 758-773.

Beecham, S., Hall, T. and Rainer, A. 2003a. Defining a Requirements Process Improvement Model. University of Hertfordshire, Hatfield. Technical Report TR 379.

Beecham, S., Hall, T. and Rainer, A. 2003b. Software Process Improvement Problems in 12 Software Companies: An Empirical Analysis. Empirical Software Engineering, 8 (1): 7-42.

Brodman, J. G. and Johnson, D. L. 1994. What Small Businesses and Small Organisations Say About the CMM. 16th International Conference on Software Engineering, May 16-21, Sorrento, Italy. IEEE. 331-340.

Christie, A. M. 1999. Simulation in support of CMM-based process improvement. Journal of Systems and Software, (46): 107-112.

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.

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. 1982. Quality, Productivity, and Competitive Positition. Massachusetts Institute of Technology Center for Advanced Engineering Study, Cambridge, MA.

Dorfman, M. and Thayer, R. H. 1997. Software Engineering. Los Alamitos, CA: IEEE Computer Society Press.

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., 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.

Ferraiolo, *K. M.* 2002. *Considerations for Re-architecting the SEI CMM Building Engineering CMMs*: http://www.systemcorp.com/2002/downloads/frames/karen_frame.html. Accessed *17.6.02*.

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.

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.

Hall, T., Beecham, S. and Rainer, A. 2002a. Requirements Problems in Twelve Companies: An Empirical Analysis. EASE, Spring, Keele.

Hall, T., Beecham, S. and Rainer, A. 2002b. Requirements Problems in Twelve Companies: An Empirical Analysis. IEE Proceedings for Software, October, 149: No.5: 153-160.

Hofmann, H. F. and Lehner, F. 2001. Requirements Engineering as a Success Factor in Software Projects. IEEE Software,: 58-66.

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

Kitchenham, B. A., Hughes, R. T. and Linkman, S. G. 2001a. Modeling Software Measurement Data. IEEE Transactions on Software Engineering, 27 (9): 788-804.

Kitchenham, B. A., Pickard, L., Linkman, S. G. and Jones, P. 2001b. A Preliminary RIsk-based Software Bidding Model. Keele University, Keele. Technical Report TR0103.

Kitchenham, B. A., Pickard, L., Linkman, S. G. and Jones, P. 2002. A Framework for Evaluating a Software Bidding Model. Proceedings of the Conference on Empirical Assessment in Software Engineering, 8th - 10th April 2002, Keele University.

Koubarakis, M. and Plexousakis, D. 2002. A formal framework for business process modelling and design. Information Systems, 27: 299-319.

Krogstie, J. 1998. Integrating the Understanding of Quality in the Requirements Specification and Conceptual Modeling. ACM SIGSOFT, 23 (1): 86-91.

Lauesen, S. and Vinter, O. 2001. Preventing Requirement Defects: An Experiment in Process Improvement. Requirements Engineering Journal, 6: 37-50.

Loconsole, A. 2001. Measuring the requirements management key process area. Proc. of the ESCOM 2001, April, London. 67-76.

Madhavji, N. H. 1991. The process cycle. Software Engineering Journal, September: 234-242.

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.

McFeeley, B. 1996. IDEAL: A User's Guide for Software Process Improvement. Software Engineering Instutute, Carnegie Mellon University, Pittsburgh, PE, USA. CMU/SEI-96-HB-001.

Neissink, F., Clerc, V. and Vliet, H. 2002. The IT Service Capability Maturity Model. Software Engineering Research Centre, Utrecht. http://www.itservicecmm.org

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.

Perry, D. E., Staudenmayer, N. A. and Votta, L. G. 1994. People, Organizations and Process Improvement. IEEE Software, 11 (4): 36-45.

Pfleeger, S. L. and Rombach, H. D. 1994. Measurement based Process Improvement. IEEE Software, (July): 9-11.

Potter, N. and Sakry, M. 2001. Practical CMM: Software Development. Accessed 9.27.02. http://www.sdmagazine.com/print/documentID=11079

Pressman, R. 2001. Software Engineering 5th Edition: McGraw Hill.

Rainer, A. and Hall, T. 2002. Key success factors for implementing software process improvement: a maturity-based analysis. Journal of Systems and Software.

Reifer, D. J. 2000. The CMMI: it's formidable. Journal of Systems and Software, 50: 97-98.

Rombach, H. D. 1990. Software Specifications: A Framework. in Standards, Guidelines, and Examples on System and Software Requirements Engineering. Dorfman, M. and Thayer, R. H., eds.: IEEE Computer Society Press Los Alamitos, California.**:** 368-408.

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, June 22-24, Oulu, Finland. Technical Research Centre of Finland ESPOO 1999. 508-522.

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).

Sawyer, P., Sommerville, I. and Viller, S. 1999. Capturing the Benefits of Requirements Engineering. IEEE Software, 16 (2): 78-85.

Smith, R. C. 1998. Using a Quality Model Framework to Strengthen the Requirements Bridge. International Conference on Requirements Engineering (ICRE '98), April 06 - 10, 1998, Colorado Springs, CO. IEEE. 118-125.

Sommerville, I. and Sawyer, P. 1997. Requirements Engineering A good practice guide. Chichester: John Wiley & Sons Ltd.

Appendix 1: Success Criteria

Objectives 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 (Madhavji, 1991) . The Table below is set in alphabetical order, not in order of importance.

Objective	Purpose	Rule	References
Adherence to CMM characteristics	"Maturity levels help characterise a process and set out a strategy for its improvement". The concept of the CMM is that you cannot proceed to the next level of maturity without having the previous level's processes in place.	a) The model must be systematic and sequential to take on the CMM characteristics The requirements model must adhere to CMM maturity concepts	(Sawyer *et al.*, 1999).
	The Requirements CMM is to be used in conjunction with the SEI's Software CMM, particularly as it is being used by many companies	b) all 5 maturity levels should be consistent in concept – it should be possible to use the Requirements CMM when using a Software CMM model for process improvement	
	This logical pattern of building on firm foundations is followed in the Requirements CMM – as the fundamental organizational and management elements apply to any organisation"	c) The requirements model must mirror the CMM and extend the given requirements processes	(*Ferraiolo*, 2002)
	The new model should be recognisable as a CMM offshoot to allow for requirements processes to link to software development processes	d) The model should be recognisable as a CMM offshoot	
	It will help with usability if the Requirements CMM can be viewed in isolation without the need to refer to the parent Software CMM model.	e) It should be possible to view and use the requirements model independently	
Assumptions	Listing assumptions will help establish a common understanding of the purpose of the model; who the model is aimed at; and what the user is expected to know about the model.	a)That a defined, documented, executed and measured requirements process is a prerequisite for effective improvement	(Rossi, 1999)
	There is little point in having a mature requirements process if all the other processes are at a lower level of maturity.	b) The requirements model will be used in conjunction with other software process improvement models such as the Software CMM, ISO/IEC 155402 or CMMI.	
	We do not want to re-invent the wheel, i.e. repeat what is given in the CMM in terms of good practice and methods.	c) The user/reader will be familiar with the concepts of the CMM	
	It is not possible to incorporate how to implement the improvement activities, we presume that an on-going improvement programme will include issues such as how best to involve practitioners.	d) There is an assumption that Software Process Improvement concepts are known and are being implemented	(McFeeley, 1996) section 1.2
Boundaries	Model boundaries should be clearly defined (i.e. define what the model will include and exclude). We take this requirement from the field of formal process modelling where the model must show constraints and describe limiting factors.	a) The model will define the properties of requirement processes at 5 levels of maturity.	(Koubarakis and Plexousakis, 2002).
		b) The model will encompass activities relating to requirements engineering to include both technical and organisational processes.	(Hall et al, 2002).
	"The CMM does not describe all the process areas that are involved with developing and maintaining software. Only those process areas identified as key determinants of process capability have been described in the CMM".	c) Processes will be included in the model in terms of priority and logical importance, not in terms of completeness. I.e. the model will not list every individual process involved in requirements engineering.	Paulk et al, 1995.
	Risk assessment is viewed as a process that the requirements process will feed into, as is costing, delivery dates and bidding processes, design and other software development processes. These processes are likely to be dependent on the requirements process.	d) The model will not explicitly model processes involved in assessing risk, costs, delivery dates and bidding processes. The requirements processes however should be compatible with these other processes	(Kitchenham *et al.*, 2001b)
	Due to time constraints the list of processes may not be complete as they cannot be tested in a project environment as this would span several years	e) Key determinants of requirements process capability will be listed based on the evidence of our empirical findings, the CMM and the requirements literature.	Hall et al, 2002; Beecham et al, 2003.

	We do not cover how to motivate practitioners into using implementing best practices. such as, 'Identify Business Needs and Drivers for Improvement".	f) methods for motivating practitioners to buy into Software Process Improvement activities are outside the scope of this work.	(McFeeley, 1996) section 1.2;(Baddoo and Hall, 2002; Rainer and Hall, 2002)
Clarity	The CMM has been criticised for having unclear vocabulary. We address this problem by limiting the use of acronyms and creating a glossary for clarification of terms used in the model.	A glossary of terms is included for clarification (TR 379 "Defining a Model of Requirements Process Improvement").	(Brodman and Johnson, 1994).
Consistency	In order to understand the model it is important that there is a common language . In software engineering (as in many other disciplines) there can be many interpretations of what appears to be a straight forward word.	Where possible the language must be consistent with the CMM. However, the language should be kept simple and acronyms avoided whenever possible	(Smith, 1998, (section 3.2))
Detail	The level of detail provided in the model should allow for expert evaluation and validation It should be concise and clear. A criticism of the new Integrated CMM is that is so detailed that no one will want to wade through all the material, "It's size alone makes it hard to comprehend and use for process improvement"	The new model should be sufficiently detailed to allow experts to form opinions on its usefulness and correctness in fulfilling the above criteria without overwhelming them with information.	(Reifer, 2000).
Ease of Use	Usability is a key requirement of any process improvement model	The model should be easy to use (requiring little/no formal training).	(Sawyer et al, 1999)
Extendable	The model should allow users to question and investigate their current practices as a first step to requirements process improvement	provide a platform for practitioners and researchers to investigate methods for improving the requirements process	
Focus	To help practitioners recognise processes that are key to improving the requirements phase in software development	The model should describe key requirements processes	
Glossary	Defining terms is essential for understanding, possible replication and model enhancement. There is a danger that definitions become context dependent leading to inconsistency in interpretation.	All terms used in the model will be clearly defined. A list of acronyms is included in the glossary	(Kitchenham *et al.*, 2001a). (Rombach, 1990)
Granularity	Having a consistent level of granularity allows for prioritisation of processes and ensures that there is no repetition.	Each stage of development should describe processes at similar levels of granularity Level of granularity must be consistent	(Kitchenham *et al.*, 2001a)
Tailorable	Model development may involve some changes and additions to the process as a result of expert feedback and individual application needs	The model should be tailorable	Paulk et al, 1995
Understandable	Understanding is a prerequisite for effective process improvement and management	The model should be easy to understand.	(Rossi, 1999)
Uniqueness	Processes described at different levels of granularity within one level could result in a process being defined individually, while also being part of a higher level process	Each process should have only one occurrence within each Level	
Verifiable	To assess whether a process is useful, well implemented the criteria needs to be verifiable	The model should be verifiable	(Kitchenham *et al.*, 2002)

Appendix 2: Problems raised by our empirical study by maturity level:

Table 2: Total CMM Level Probs Normalised

	CMM 1	CMM 2	CMM 3	CMM 4
Org problems	139	96	92	90
Tech Probs	73	81	62	6
Total	212	177	154	96

Table 3: 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	139	32	46	15	232

Table 4: REQUIREMENTS PROCESS 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. Inadequate 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	73	27	31	1	132

Appendix 3: Two examples of a Requirements CMM approach to process improvement

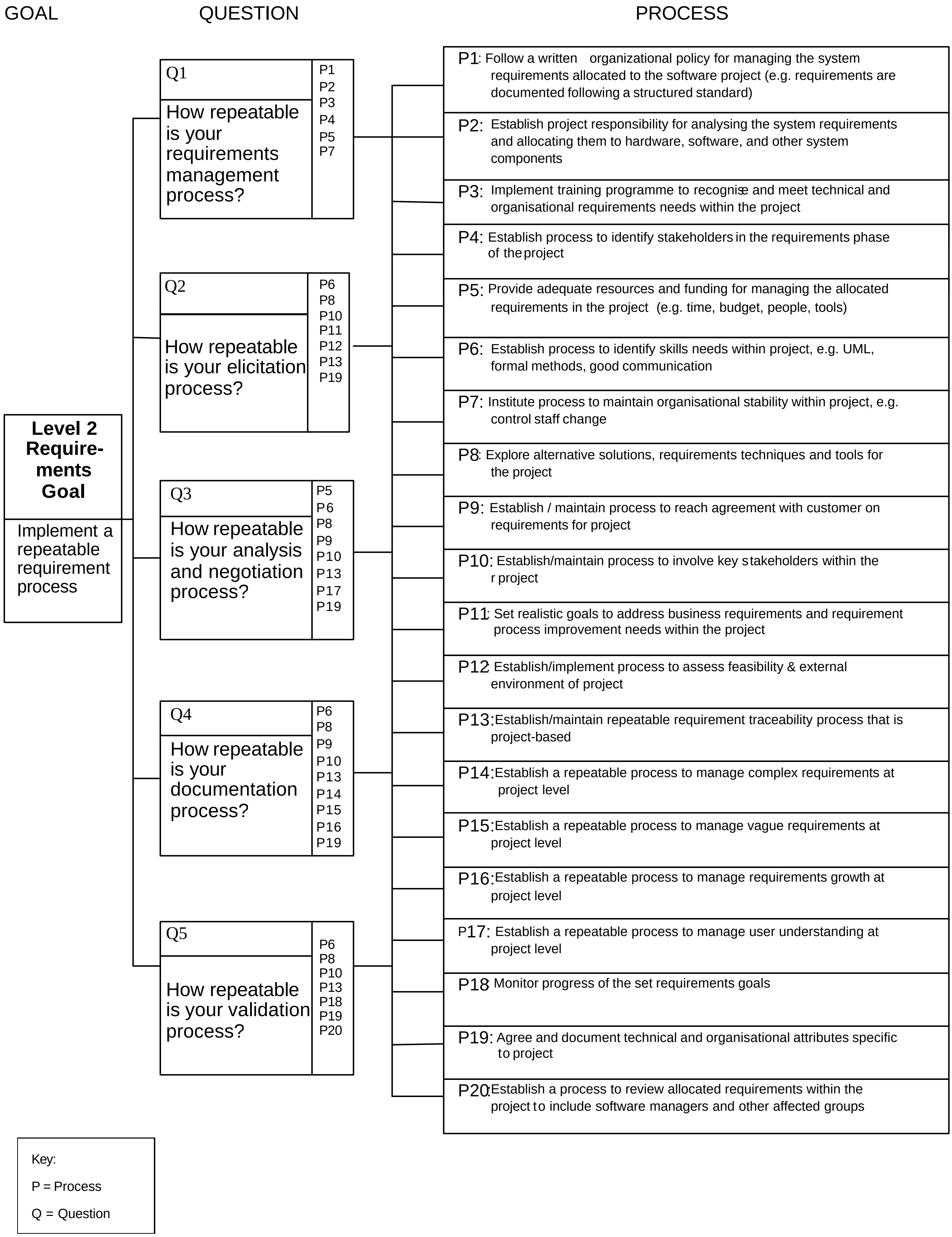

Figure 6: Level 2 goal-focussed requirements processes – A high level analysis

Establishing a repeatable "Identify Stakeholder' process at a project level – derived from Process P4 in Figure 6.

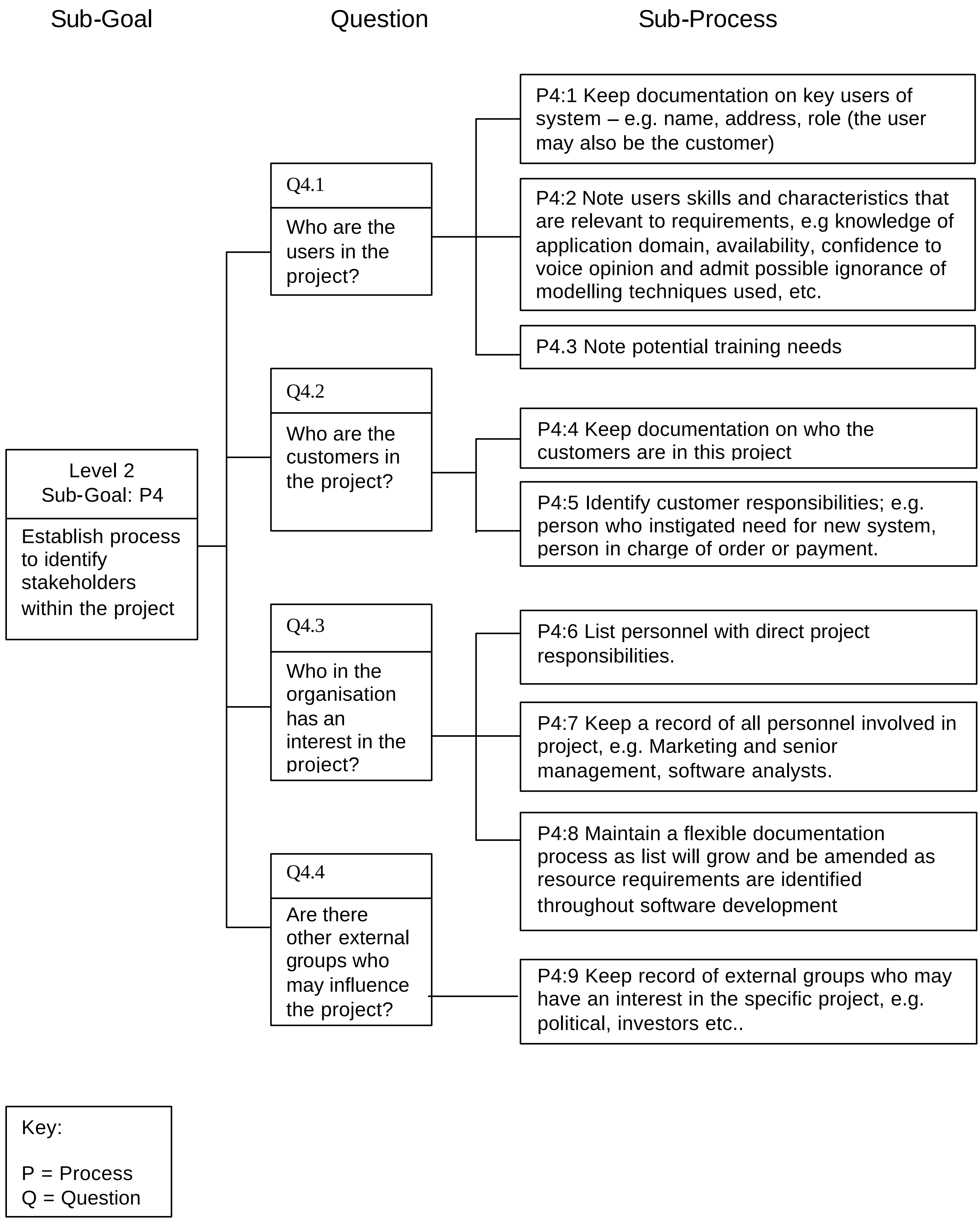

Figure 7: Example of a Level 2 Requirements Process Guideline – A finer grained analysis