A Method for Assessing Requirements Engineering Process Maturity in Software Projects

Tony Gorschek
Kaarina Tejle

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden
This thesis is submitted to the Department of Software Engineering and Computer Science at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 40 weeks of full time studies.

**Contact Information:**
Authors:
Tony Gorschek  
E-mail: tbg@home.se

Kaarina Tejle  
E-mail: kaarina.tejle@home.se

University advisor:
Mikael Svahnberg  
Department of Software Engineering and Computer Science
Abstract

The area of Requirements Engineering is often underestimated in value in the area of Software Engineering. According to certain sources the failure rate of IT investments is over 60%. In addition problems introduced through the Requirements Engineering of a project accounts for something like 50% of the total debugging costs. The main reason for this is a low level of maturity pertaining to the Requirements Engineering process.

This thesis introduces a model that can help organizations improve their Requirements Engineering process. A first step in process improvement is process evaluation. The REPM model has the purpose of measuring the maturity level of the Requirements Engineering process in projects, and to give a basis for what steps to take in order to improve on it. In addition to the model a method for using the model is introduced. The model and method are subsequently designed, implemented and validated.

The validation takes the form of interviews and case-studies in industry featuring four companies and four projects of varying size. The project evaluations were conducted on-site in both Sweden and in Ireland.

It is shown that the REPM model in combination with the method is a good way to evaluate the Requirement Engineering process of a project. It gives a picture of the current state of the Requirements Engineering process in a project and, more importantly, how the results of the evaluation can be used for process improvement.

Keywords: Requirements Engineering, Process Evaluation, Process Improvement.
# Table of contents

1 Chapter One – Introduction

1.1 History

1.1.1 Computers and Software

1.1.2 Requirements Engineering

1.1.3 Knowledge Engineering

1.2 Requirements Engineering – An Introduction

1.2.1 The Requirements Engineering Process

1.3 Requirements Engineering - Problems

1.4 Our theory

1.4.1 The Construction of the REPM Model

1.4.2 REPM Model – An Introduction

1.5 Literature and Related Work

1.6 Disposition

1.6.1 Part One

1.6.2 Part Two

1.7 Acknowledgements

2 Chapter Two – The REPM Model

2.1 Structure of the REPM Model

2.1.1 Notation

2.1.1.1 Optional Actions

2.2 Requirements Engineering Process Maturity

2.2.1 REPM 1 – Initial (Wood)

2.2.2 REPM 2 – Basic (Bronze)

2.2.3 REPM 3 – Formulated (Silver)

2.2.4 REPM 4 – Developed (Gold)

2.2.5 REPM 5 – Advanced (Platinum)

2.2.6 Adding to the model

2.3 Usage of the REPM model

2.3.1 Project Evaluation

2.3.2 The Result

2.3.2.1 Satisfied-Explained Actions

2.3.2.2 Result Presentation through Diagrams

2.3.3 Result Presentation – an Example

2.3.3.1 Action Summary

2.3.3.2 MPA of Requirements Elicitation

2.3.3.3 MPA of Requirements Analysis and Negotiation

2.3.3.4 MPA of Requirements Management

2.4 Conclusion

3 Chapter Three – REPM Model Validation

3.1 Validation Method

3.1.1 Selecting Interview Subjects

3.1.2 Interview Questions

3.1.3 Presenting the Interview subject
Chapter One – Introduction

The purpose of this chapter is to give an introduction to the development of software engineering in the context of Requirements Engineering from a historical perspective, as well as give an introduction to Requirements Engineering and the process. Some of the problems in the area of Requirements Engineering are also presented and a discussion of their causes. We also give the reader an introduction to the REPM model, why it was created and where the inspiration for creating it came from.

This chapter is divided into seven main parts, a historical perspective of Computers and Software, Requirements Engineering and Knowledge Engineering, an introduction to the discipline Requirements Engineering and a formulation of some of the problems facing the field. Our proposal for how some of the problems can be alleviated using the Requirements Engineering Process Maturity (REPM) model, and an introduction to that model and how it was created. Last in this chapter we have information about literature and related work that has had an impact on the creation of the REPM model.

1.1 History

1.1.1 Computers and Software

The history of Software Engineering can according to Robert Glass be divided into three different periods: The Pioneering Era (1955-1965), The Stabilization Era (1965-1980) and The Micro Era (1980- Present) [1].

During the Pioneering Era the most important development was the fact that new computers where distributed very frequently. New computers would appear on the market every year or two making it necessary to rewrite the software programs for each new machine. The programmers themselves did not have computers of their own but had to go to the ‘machine room’ and reserve time “to program”, which in turn gave rise to the problem that it was almost impossible to predict a projects status and finish date. A consequence of the frequently distribution of new computers and the need to translate and update the old software was that several high-order languages where developed, e.g. FORTRAN, COBOL and ALGOL. The notion of reuse flourished as time spent in the “computer room” was expensive and old code (solutions) were reused. The idea was that the hardware was the main thing; software was more of less a necessary nuisance.

The Stabilization Era is said to have come when the IBM 360 (1964) was first introduced. At that time the worst problem that had come out of the job-queue system was the enormous bureaucracy grown around the central computer system. The result of this bureaucracy was the problem with major turnaround time. The IBM 360 was the largest software project to date and the software programmers could finally spend time programming and writing new code rather than having to reuse old. The 360 also combined the different applications into one machine taking away the problem of needing different machines for different applications, e.g. one for scientific and one for business
applications. This also made the separation between the different application people diminish which led to an impact on the sociology of the field [1].

Demand for programmers on the job market exceeded the supply. This and the fact that software became a corporate asset lead to the emerging of academic computing disciplines in the late 60’s. However the software engineering discipline did not yet exist.

In the middle of this era the notion of structured programming was becoming accepted. The control was taken by the standards organizations that realized that the vendor who defined the standards could gain significant competitive advantage by making the standards match their own technology [1]. The hardware-centered view changed somewhat during this era – shifting towards the realization of the value of software. Two Algol 60 men, Friedrich Bauer and Louis Bolliet, had just coined the term “software engineering”.

The beginning of the Micro Era was signaled with the possibility for every programmer to have his/her own computer at his or her desk. Looking at the programming field itself the major change was that user-friendlier GUI replaced the old JCL. This in combination with the PC revolution gave rise to the use of computers and software by people outside of the engineering disciplines. Not to mention that companies in every field started to use computers in almost all areas from aiding research to managing payrolls [1].

In the early years of computers the hardware was seen as the important factor and the development was driven by large organizations like the US Department of Defense. Computers were for a select few.

As time progressed and computers became less exclusive the focus was gradually turned to software. The main force behind the computer and software development migrated from large organizations, Department of Defense (DoD) and large corporations, to companies of all sorts, and last but not least people sitting at home or in the office using their PC in everyday work - using software.

1.1.2 Requirements Engineering

A historical problem within system development is the fact that the users where considered to be more of a nuisance than to have a part in the system. An example of this was documented in 1971 by Harold Sackman in “Mass Information Utilities and Social Excellence” and even as far as in the late 1980’s when Microsoft first started to conduct usability studies and they found that 6 to 8 out of 10 users could not understand the user interface and get most features, the response from the programmers was “Where did they find eight dumb users?” [2].

A major problem in the 1970’s was that computer manufacturing was very costly, the software (systems) ended up being of a complex nature and the question of problem definition was never considered to be important. This made the developers and the technically minded specialist, that knew how to build and maintain large and complex systems, into a kind of priesthood.
Already in the early 1970’s the Ethnological Approach, the use of ethnographic methods to identify objects in the social environment for the system to be developed, was presented. But it was not until the early 1990’s that it was explored enough to be taken more seriously. The thought that user-centered design, where users participate in the interface design process from an early stage in the process, was likely to lead to more usable user interfaces.

The introduction of SIMULA, and in essence Object Orientation, in the 1960s by Ole-Johan Dahl and Kristen Nygaard at the Norwegian Computing Centre (NCC) made it possible to change in what way a system and its parts were perceived. Later Firesmith wrote the paper ‘Structured Analysis and Object-Oriented Development are not compatible’, where he argued that is was better to describe the system as a set of objects, identifying interfaces between such objects rather than between functions. The approach broke down user requirements and was considered to be an important step towards modern system development. One can see that there might have been concern as to how object-oriented analysis could ever be extended to cover the start of the system life cycle, the user requirements phase.

The Taylorian time-and-motion studies in the early 20th century mass-production factories first presented the idea of using scenarios. It is possible that a reaction against the Taylorism made the use of scenarios in discovering and prioritizing requirements delayed to a more recent date. “Object-Oriented Software Engineering A Use Case Approach” from 1992 by I Jacobson is one of the first books completely devoted to the subject.

The scenario approach made developers more aware of the fact that end-users, and others affected by the system, had to be taken far more seriously as an important part of the process. When developing systems that are intended to be used by people in the varied contexts of their work, private, social and leisure activities the focus of the design must be on the suitability of the designed artifact to support and complement human activity. In short the stakeholder, i.e. people using or otherwise affected by the system, finally started to be in focus. Around the same time the developers realized that the requirements needed to be expressed in natural language to help the users to see if the requirements are the ones needed and wanted.

In 1996 the article ‘When Good Enough is Best’ by E. Yourdon was published. The article concerns the huge problem of specifying and building systems in the face of constant change. He argues convincingly for Requirements Management to implement the act of prioritizing the requirements and the fact that not everything can be done in limited time and within budget.

Starting from the 1990’s Requirements Engineering had emerged into its own field of study, this being witnessed by the commencement of two series of international meetings. One being the establishment of an international journal, ‘Requirements Engineering Journal’ published by Springer, and the other a conference and symposium held in alternating years sponsored by IEEE. In the late 1990’s the field had grown into being...
able to support several smaller conferences and meetings discussing Requirements Engineering.

In summary one could say that during the late 1990’s three main ideas took form. First the realization that modeling and analysis could not be performed isolated from the organizational and social environment in which the system should later be placed. The second one being the notion that Requirements Engineering should not focus on specifying the functionality of a system but instead concentrate on the properties of the environment. Only by describing the environment and what the system should accomplish in that environment is it possible to ascertain the validity of the system. This can be achieved through scenarios and modeling stakeholder goals. The third and last was that Requirements Engineering should take seriously the need to analyze and resolve conflicting requirements, to support stakeholder negotiation, and to reason with models that contain inconsistencies, since rapid changes in requirements were a reality the attempt to build consistent and complete models was futile [19].

The major trends that have driven the status of Requirements Engineering are falling price and increased accessibility of computer-based systems, growing interactivity bringing a widening range of users with rising expectations and rising size, and cost of system failures despite ever-better development tools. Delivered functionality that the customer does not need and did not ask for is also fairly usual. These trends have forced system and Requirements Engineering to be transformed into full engineering disciplines [3].

1.1.3 Knowledge Engineering

The field of knowledge engineering was established in the 1970’s and has from then on evolved to be used in many fields of which business administration is one. Knowledge engineering is good for finding bottlenecks and opportunities, e.g. how organizations develop, distribute and apply their knowledge resources. It is also a provider of methods to understand the processes and structures used by knowledge workers. The main goal of knowledge engineering is to enable the possibility for building better knowledge systems [27].

The history of knowledge systems started around 1965 when the general-purpose search engine (GPS) was built. In 1975 the first generation rule based systems was developed. The emerging of structured methods for project development came around 1985. In recent years (1995) the mature methodologies of CommonKads have been developed and implemented. CommonKads can be described as guidelines for developing large scale knowledge systems in a structured, controllable and repeatable way. The work with the CommonKads was started as early as 1983 and at that time there was little interest for methodological issues.

In traditional views knowledge engineering is seen as a process of extracting/finding the knowledge of how the system should be and forming this knowledge in computational form to a machine. Today Knowledge Engineering is seen as a modeling activity that makes it possible to highlight certain aspects of the system and leave others aside. The
difference between software developers and knowledge engineers is that the software developer takes the system as a reference point in the middle of their design and analysis activities, while the knowledge engineers studies the domain, humans around and the users to develop the system.

The area of knowledge engineering is not part of the main stream research areas but one has to note certain things: Parts of knowledge engineering is also found in Requirements Engineering. Domain knowledge, humans and user surrounding the system are considered in both disciplines [27]. It is important to acknowledge knowledge engineering in terms of having addressed many of the areas now being addressed in Requirements Engineering.

1.2 Requirements Engineering – An Introduction

There are many definitions of Requirements Engineering, Somerville (1992) states that requirements capture and analysis is “the process of establishing the services the system should provide and the constraints under it must operate” [4, p. 1]. Goguen says that “requirements are properties that a system should have in order to succeed in the environment in which it will be used” [5, p. 17]. Thus, it could be concluded that Requirements Engineering is the engineering of these properties.

Requirements Engineering is a multi-disciplinary, human-centered process where the requirements engineer may be expected to master skills from within a different number of areas [9]. The tools and techniques used within Requirements Engineering draw upon a variety of disciplines, e.g. the areas of characterizing systems, system analysis and sociology. Requirements Engineering has to span the gap between the informal world of stakeholders and their needs, to the more formal world of software behavior [6]. In short one could say that Requirements Engineering covers all of the activities involved in discovering (sometimes called catching), documenting and maintaining a set of requirements for a computer-based system.

The term stakeholder is an important one. This group is comprised of all people that have a direct or indirect influence on the system requirements, e.g. system end-users, people involved in processes that are influenced by the system, engineers producing and managing the system and external bodies such as regulators [7]. Without the stakeholders there would be no system, or need for one for that matter. The official statement of the system requirements, discovered through amongst other sources the stakeholders, is called a requirements document.

1.2.1 The Requirements Engineering Process

A Requirements Engineering process consists of a structured set of activities, which are followed to derive, validate and maintain a system’s requirements document [7].

A complete description of a Requirements Engineering process should include what is done at what time, who performs a certain activity, what resources are allocated, who is
Chapter One – Introduction

responsible for what activity, what the result of each activity is and what tools are used to support the Requirements Engineering process [7] [9].

The standardization of processes is a way to ensure repeatability and in many cases also a part of quality assurance. Standards like ISO9000 are commonly used when it comes to software engineering but does not include a framework for the Requirements Engineering process. Very few organizations have a clear cut and standardized (defined and repeatable) Requirements Engineering process [7] [9] – at best there is a definition of the result of the process (process outputs), i.e. the requirements document and how it should be structured. An example of such a standard is the IEEE/ANSI 830-1993 [12]. Requirements Engineering processes at different organizations can be separated by looking at certain key factors. Technical maturity, disciplinary involvement, organizational culture and application domain [9] are some of them.

Based on literature, we see three main activities (Main Processes) in a Requirements Engineering process which should be present, called Elicitation, Analysis and Negotiation and Management [6] [9] [14] [15] [16].

Requirements Elicitation

Requirements Elicitation is the activity mostly regarded as the first step in the Requirements Engineering process [6]. Briefly one could say that it is the work of finding and revealing the requirements from the stakeholders, system documents, domain knowledge and market studies. The term “elicitation” is used instead of e.g. “capture” to avoid the notion that the requirements are just lying around waiting to be collected instead of the fact that you have to reveal them through work and investigation. The elicitation activities are closely related to the rest of the Requirements Engineering process. Within the elicitation area there are many elicitation techniques. Below, we present a short description of some of them:

Conventional techniques include the use of interviews, surveys and questionnaires to gather data from stakeholders. In addition analysis of existing documentation e.g. process models, organizational charts, standards and manuals for existing systems [6] can be conducted. Scenarios and use cases can be a very effective way to concretize abstract descriptions into real-life examples helping in the data gathering from stakeholders [9].

Observation (Contextual techniques) can be a good way of eliciting how things are done in contrast to asking stakeholders to describe what is done and how. It can be difficult for stakeholders to articulate fairly simple routines and tasks [9] [6]. Furthermore the context and environment in which the stakeholder works can be of importance – contextual approaches are context driven.

Reuse is basically the process of using existing knowledge when developing a new system, maybe certain things (requirements) have already been described and/or implemented.

Group Elicitation techniques such as brainstorming, Joint Application Design (JAD) workshops can also be valuable for elicitation.
Chapter One – Introduction

Requirements Analysis and Negotiation

Requirements Analysis and Negotiation denotes that the requirements are analyzed in detail. If the requirements are found to be relevant and accurate a formal negotiation follows where stakeholders are the ones approving which requirements are to be accepted. The work of analyzing the requirements is done with the intention to find possible conflicts, overlaps, omissions or inconsistencies – the interaction of requirements is also an important thing to study [9]. The requirements can be classified and grouped according to everything from functionality to importance. The risks are assessed for the requirements (individually or groups) in order to build an understanding of the potential problems.

The requirements are examined at an early stage and unrealistic requirements can be sorted out. The work of sorting out the requirements can also be mentioned as the negotiation part – basically different stakeholders have different views of what is important. Different stakeholders also have different power over the decisions being made [9] [11]. Another activity here is usually to prioritize the requirements.

Requirements Management

Requirements Management is an ongoing process during the entire life-cycle of the software project. It starts at the very beginning with the management of writing down the elicited requirements and is a part of the process until the very end of the last documentation is finished, i.e. when the software project is terminated. The way the requirements are documented and managed plays an important role in the way it ensures that they can be easily read, analyzed, rewritten and later validated. Important parts of the Requirements Management process are:

Requirements identification and storage is basically the notion that every requirement should be uniquely identified and stored in the requirements document and/or in a database [9] [7].

The requirements document is a document that communicates the requirements to the customers, system users, managers and system developers. Many things are recorded here and linked to the requirements, e.g. requirements rationale. To achieve the readability that is necessary for the requirements a variety of documentation standards have been developed. All the standards provide clear guidelines for structuring the different relevant documents. The requirements document is basically a gathering of all the documentation produced during the Requirements Engineering process.

Requirement change policy should consist of the information of how to manage a change of a requirement, how the change should be proposed, analyzed and reviewed. All this to simplify and effectively change the requirements in the way the stakeholder or other persons involved wants them changed [9] [7].

Traceability is important in an environment that is prone to change. The impact of change on the rest of the system must be known [9] [7]. Traceability is the ‘ability to describe and follow the life of a requirement in both forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and
use, and through all periods of on-going refinement and iteration in any of these phases)"
[13 p. 94].

1.3 Requirements Engineering - Problems

The area of Requirements Engineering is often underestimated in value in the area of Software Engineering. According to certain sources the failure rate of IT investments is over 60% [17]. In addition problems introduced through the Requirements Engineering of a project accounts for something like 50% of the total debugging costs [4]. One of the major causes for this is the lack of a complete and/or adequate requirements specification [7] [4] [16]. The requirements specification is a direct result of the Requirements Engineering process and it stands to reason that an inadequate specification is a result of a Requirements Engineering process with a low maturity level [9].

Why then is there such a high cost derived from inadequacies in Requirements Engineering? There is no simple answer; one thing that could be a contributing factor is that engineers (people in general for that matter) seem to look at solutions to problems instead of specifying requirements in an adequate way [8]. Another contributing factor to the problems with Requirements Engineering is assumption of domain. Requirements are located in the application domain but most effort is put into discussing and describing the machine, which is the solution [8]. Understanding the domain, its limitations, benefits and effects on the system is crucial – if you can not understand the environment how are you to build a system that can interact and add value to it?

One could argue that the problems mentioned above are symptoms of an inadequate Requirements Engineering process. When a process in an organization is standardized the process obtains a level of maturity. The level of maturity can be thought of as to the extent the organization has defined controls and actively supports the process [9].

Looking at process evaluations today there are a few models for how to develop and measure processes. One of them is the Capability Maturity Model, CMM. CMM can be described as a common sense application of process management and quality improvement concepts to software development and maintenance but it focuses on software development and does not cover the Requirements Engineering process [7] [9]. ISO 9000 is another standard for quality management processes and it has been adapted in different variants to suite software development, e.g. The TickIT Guide 2001. The main thing is that CMM and ISO 9000 does not say much about Requirements Engineering and subsequently little about how the quality of the Requirements Engineering process should be maintained and ensured. Many companies and other organization use CMM or ISO 9000 and are satisfied – this results in that their adapted "quality assurance standard/framework" does not help them in the area of Requirements Engineering. Without a standard for ensuring the quality of the Requirements Engineering process it is hard to ensure the result of the Requirements Engineering process. A consequence of this can be that requirements do not reflect the real needs of the customer of the system, requirements are inconsistent and/or incomplete and
Chapter One – Introduction

requirements (with pertinent information, See section 1.2.1) are not specified in a standardized and adequate manner.

There are several ways in which one can quickly evaluate if the current Requirements Engineering process is adequate, one could pose questions like; is the Requirements Engineering process usually over budget, is the requirements document understandable and is there a lot of rework stemming from requirements errors [7]?

1.4 Our theory

Looking at methods for process quality measurement and improvement they tend to cover the area of Requirements Engineering poorly. An example of this is CMM. We feel that there is something missing from the very extensive, large model, i.e. it covers the area of Requirements Engineering inadequately. In essence it omits the Requirements Engineering process altogether, other than having certain activities (called Goals) mentioned such as “requirements are agreed to by all affected groups” [18 p.62]. CMM does not cover how the quality of the Requirements Engineering process should be secured or what activities should be present for the Requirements Engineering process to achieve a certain maturity level.

We have found that it is not easy to assess the maturity of a Requirements Engineering process for a certain project, and subsequently it is difficult to know what is missing and what could be done to improve the process. A model and a method for measuring the maturity level of the Requirements Engineering process within a project can help organizations to identify weaknesses and improve their Requirements Engineering process.

We propose a model for assessing the Requirements Engineering process within software engineering projects. This model should cover the area of Requirements Engineering, i.e. what should and/or could be done during the Requirements Engineering process. The model can be used to evaluate the Requirement Engineering Process Maturity (REPM) for a certain project, i.e. how mature the Requirements Engineering process of a certain project is (See section 1.4.2). A spin-off effect of this is that the model basically gives the user, i.e. the project evaluator, a blueprint for how to improve the Requirements Engineering process.

In addition to the model we also propose a method for using the REPM Model for project evaluation. The method consists of several parts; a type of user manual to the model, a checklist to follow when doing a project evaluation and help with how the results from the model should be evaluated to draw the best benefit from the model.

1.4.1 The Construction of the REPM Model

The basic construction of the REPM Model was a result of literature research in the field of Requirements Engineering (See section 1.5), and the study of models like CMM and ISO 9000. The construction of a model is however just the first step. The second step is
to validate the model through an interview (See chapter 3). The interviewee is picked for his extensive experience in industry in both the fields of Requirements Engineering and Software Engineering in general. The purpose of this first validation is to scrutinize the model and judge its applicability to the Requirements Engineering processes in industry. This is done to see what members of industry think about the model and to get their feedback on what is lacking or, in their opinion, needs to be revised.

The third step is to test the model (See chapter 4). A method for using the model on projects is constructed and subsequently the REPM model is used to test four projects from industry. The idea is to evaluate the model’s usability and to gather data for model improvement at the same time. In addition to testing the model itself the method for using the model is validated.

1.4.2 REPM Model – An Introduction

The REPM model is basically a map of sorts, describing the Requirements Engineering process and its constituents. There are three main areas of activity; Elicitation, Analysis and Negotiation and Management (See section 1.2.1) – these are called Main Process Areas (MPAs) in the model. Under each of these there are a number of sub-areas called Sub Process Areas (SPAs) and ultimately at the bottom there are Actions – describing what could and/or should be present. Basically you get a tree-structure where MPAs are the top nodes and the Actions are the bottom ones. SPAs are a way to further granulate the MPAs into different categories.

The reason for the structure is that it enables the models content to be arranged in a way that does not hinder further development of the model, e.g. you can add to the model or move Actions up in the structure effectively converting it into a SPA and then adding new Actions under it. A more detailed description is offered in section 2.2.6.

Every Action is mapped to a certain Requirements Engineering Process Maturity (REPM) Level spanning from 1 to 5 (See section 2.2). Our REPM Model follows the same framework as CMM (See section 1.3 and 1.5), as far as having five levels of maturity but the similarities end at an early stage. We have chosen to concentrate on the area of Requirements Engineering and not process improvement in general like CMM. In addition the REPM Model should be used for project evaluation and not evaluation of an organization’s process maturity.

The REPM Levels are important in the sense that it is possible to grade a projects Requirements Engineering process maturity. The model is a tool for evaluating on what level a project is when it comes to Requirements Engineering. Setting sensible goals for process improvement requires an understanding of the difference between an immature and a mature Requirements Engineering process, and you need a tool for measuring maturity. Continuous process improvement is based on small, evolutionary steps. These steps are described in section 2.2 as the five REPM levels.
Chapter One – Introduction

1.5 Literature and Related Work

The work of Gerald Kotonya, Ian Sommerville and Pete Sawyer has inspired the creation of the REPM model. In the book ‘Requirements Engineering – A good practice guide’ [7], Sommerville and Sawyer suggest guidelines for Requirements Engineering process improvement and formulate a step by step framework that can be followed in order to better the Requirements Engineering process.

In Sommerville’s and Kotonya’s “Requirements Engineering – Processes and Techniques” [9] a fairly good and extensive introduction to the field of Requirements Engineering is given, and this book serves as a good knowledge base reference.


Our previous experience with CMM and ISO 9000 is the main inspiration behind this thesis and the construction of a REPM model. CMM was produced by the Software Engineering Institute at Carnegie Melon University. The inspiration for the development of CMM came out of the dissatisfaction with known, consistent software problems. The development team took inspiration for the tool from among other things the success of total quality management, Crosby’s maturity grid and IBM’s process grid [10].

CMM consists of 5 levels of maturity: Initial, Repeatable, Defined, Managed and Optimizing [18] (See Figure 1.1). The first level is the Initial level where the organization has an undisciplined process - individuals manage the process, and the development of it, without a common standard for the organization. The Repeatable level is when the organization has a basic cost and schedule management procedures in place. They are at this stage likely to make budget and schedule predictions for some projects. The Defined level is depending on the fact that there is a standard software process for the organization that is constructed of the different software processes for both management and engineering and their documented activities. The fourth level is the Managed level which contains detailed measurements of both process and product quality. These measurements are collected for the purpose of controlling the process. The final level in CMM is the Optimizing level, which is built on the fact that the organization has a continuous process improvement strategy that is based on objective measurements that are in place in the organization.

The REPM model draws from all of these sources and adds to them. We use five REPM levels (just like CMM) but their meaning and content differs [See section 2.2]. Furthermore we have concentrated solely on the area of Requirements Engineering. If we look at the model presented by Kotonya, Sommerville and Sawyer we have tried to concentrate on the “bare bone” contents, i.e. only including the most important aspects in the REPM model. We have also added several parts and expanded on others. Important to notice is that one of our main objectives was to produce a model that is easy to grasp and possible to understand and use without committing vast resources to train people to be
able to evaluate projects. Our focus on projects is another thing that sets us apart from e.g. CMM, which focuses on the organization as a whole.

![CMM Five Levels of Process Maturity](image)

**Figure 1.1** CMM Five Levels of Process Maturity.

Government and private industries assess the maturity level of an organization’s software process using CMM [10]. We propose a way to assess a specific projects (not a organizations) Requirements Engineering process maturity level.

ISO 9000 embodies the idea that quality does not come from a vacuum but rather from a deliberate plan of documented activities, plans, responsibilities and the creation of a system for all of this. ISO 9000 is basically a list of “what to do”, in an ordered step-by-step scheme. This list of demands is a manual for what is to be done in order to create a Software Quality Management System. This system, or rather the organizations conformance to the “ISO-way”, is the way quality is ensured in the ISO 9000 paradigm. The first step is to evaluate the organization and the processes involved in the development process, the interaction between different parts is also important. After this every part is basically adapted to fill the requirements stated by ISO 9000. This can entail everything from “the responsibilities of the management to define, document and implement a policy for quality” to the “establishment of documented procedures for training the personnel” [20]. First we have the implementation of a system, and then all the parts are monitored and measured so that the third part, improvement, is possible. In ISO 9000 everything, every aspect and activity has to be documented [20].
Both ISO 9000 and CMM have many merits, i.e. quality assurance and process improvement. We believe that by taking what can be learnt through the work of Kotonya, Sommerville, Sawyer, Jirotka and Goguen and developing a model with primarily CMM as a guide we could make a contribution to the field of Requirements Engineering.

1.6 Disposition

The paper is divided into two main parts. The first part contains the five chapters; Introduction, The REPM Model, REPM Model Validation, Project Evaluation (REPM Model Validation Part II), and finally Conclusions and Discussion.

The second part is comprised of three appendices; Appendix I - The REPM Model, Appendix II – Interview Question REPM model validation and Appendix III – Project Evaluation Checklist.

1.6.1 Part One

Chapter 1 – Introduction
The introduction gives the reader background and historical information about Software Engineering in general, Requirements Engineering in particular, a brief description showing the connection to Knowledge Engineering, problem formulation and how our idea, i.e. the REPM model can help to alleviate some of the existing problems.

There are also sections concerning the relevant connection to literature, other process improvement/assurance models and industry.

Chapter 2 – The REPM Model
The second chapter concerns a detailed account of the REPM model pertaining to structure, contents and design.

In addition to the REPM model a method for using the model for the purpose of project evaluation is presented. A test evaluation is also presented as an example of a test case (a project evaluation) with the purpose to show how the result of a evaluation can be presented, and to give an introduction to how the project evaluations are presented in chapter 4.

Chapter 3 – REPM Model Validation
The third chapter presents the first round of validation carried out on the REPM model. The first validation was done using a single company to validate the models accuracy and content pertaining to industry.

The chapter also contains the method used for the interview, action plan for the interview (stating the demands put on the organization and subject being interviewed), result, and finally suggested changes made to the model in regards to the information gathered.
Chapter One – Introduction

Chapter 4 – Project Evaluation (REPM Model Validation Part II)
The fourth chapter contains the second validation, i.e. validation of the REPM model and
the method for using it. Four projects were evaluated. The result from each project
evaluation is presented with improvement suggestions and conclusions.

Chapter 5 – Conclusions and Discussion
The final chapter contains a discussion and the conclusions drawn from the work done in
chapter one through four.

References
A list of references to literature researched for this paper.

1.6.2 Part Two

Appendix I – The REPM Model
The entire REPM model is presented in version 1.0. A manual is also included as well as
an Action summary.

Appendix II – Interview Question
The interview questions for the first round of validation are presented.

Appendix III – Project Evaluation Checklist
The Project Evaluation checklist is presented in total.

1.7 Acknowledgements

Writing a thesis is not a solitary effort but a cooperation, or rather a collaboration,
between many parties. We would like to thank some individuals for their contribution to
this thesis.

Doctoral student Mikael Svahnberg acted as our supervisor and mentor. The support and
guidance offered by him helped us through many difficult moments and had a substantial
positive influence on the quality of work.

In addition we would also like to take this opportunity to recognize the following people;
Professor Rune Gustavsson, University lecturer Håkan Grahn and Doctoral student
Kenneth Henningsson at the Department of Software Engineering and Computer Science
at Blekinge Institute of Technology.

For all the help received during our time at Cork Institute of Technology in Ireland we
would like to thank the following people; Michelle Fornan, Jim O’Dwyer, and Aaron
Krawczyk.

We would also like to thank the companies that have participated in our research and
especially the following persons: B Sc SE Anna Hoff, Project Manager Göran Ingvarsson
(IMI), Project Manager Lars Göran Cronberg (SchlumbergerSema), Project Manager
Chapter One – Introduction

Michael O’Gorman (Vistech), Consultant Gerard O’Regan (Lecan Technologies), and Project Manager Mary Finn (Lecan Technologies).

www.sema.se

www.lecan.ie

www.vistechsoftware.com

www.im.se

www.bth.se
The purpose of this chapter is to give a deeper insight into the REPM model, its structure and constituents. Furthermore the method for using the REPM model is described. The development of the model, and subsequently this chapter, is part of our contribution. Inspiration for the model was taken from several different areas (See section 1.5) and our own experiences in the field.

Chapter two is divided into three main parts, the Requirements Engineering Process Maturity Model (described in overview in Chapter one) is at the centre of this thesis and is described in greater detail, and a detailed account of the five REPM Levels. Last we have an account of the method used for project evaluation with the REPM model and an example of a project evaluation.

2.1 Structure of the REPM Model

Designing the REPM model there is a need to categorize and organize different parts of the model into a logical and expandable structure. The main activities of **Elicitation**, **Analysis and Negotiation** and **Management** are identified (See section 1.2.1), we call them **Main Process Areas (MPAs)** – they are at the top level of the structure. Under each of these there are typically several **Sub Process Areas (SPAs)** that further granulate the MPAs into different categories. At the bottom of the tree structure we have **Actions** (Figure 2.1). MPAs are at the top, Actions at the bottom with everything from 0..n SPAs...

![General Structure of REPM model](image)
Chapter Two – The REPM Model

in-between. SPAs that are at the bottom level, i.e. have no SPAs of their own, have at least one Action. This however does not prevent SPAs that has SPAs of its own to have Actions too (Figure 2.2).

![Figure 2.2 Example of SPA levels.](image)

Actions denote an activity, e.g. to assess risks of a certain nature, and/or something that should be present, e.g. a certain document part. It is not possible to place anything (MPAs, SPAs or Actions) under an Action. If it is felt that a certain Action should be on a higher level in the model, i.e. the Action needs to be split up and described in further detail, e.g. as a result of expanding the model, the Action has to be converted to a SPA (See section 2.2.6).

A MPA may or may not have Actions of its own - Actions placed under a MPA are general in nature and are considered to be associated with the MPA and not any of the present SPAs.
Chapter Two – The REPM Model

2.1.1 Notation

Every MPA, SPA and Action has a unique identifier which denotes its place in the hierarchy (Figure 2.3). This enables tracking and when acquainted with the notation it is easy to place a certain entity (SPA and/or Action) in the structure.

![Diagram of REPM model structure]

The REPM model structure is organized into MPA, SPA, and Action levels. Each level has a unique identifier to help in tracking and placing entities within the model.

1 – Main Process Area (MPA)
There are three main Process areas. Each MPA has a unique identifier, e.g. E for Requirements Elicitation. This identifier’s purpose is to facilitate traceability throughout the model.

2 – Sub-process Area (SPA)
The identifier, E.1, denotes that the SPA belongs to the MPA “E”.

Figure 2.3 Example of REPM model structure.
Chapter Two – The REPM Model

3 – Action
Actions are the final nodes in the tree-structure that the model is comprised of. Actions do not have any sub-areas/actions. An Action is basically something to be done. An example can be the Action *E.1.a1 Ask Executive Stakeholders* which is the act of asking the executive stakeholders to identify other stakeholders. An Action can also be an item that should be obtained, e.g. *M.1.1.a1 Summary* which denotes that a summary is needed in the requirements document.

All Actions can be identified through the small letter “a” which precedes the Action number, e.g. in the case of M.1.1.a1 the “a” denotes that it is an Action (M.1.1.1 would signify a sub-process area).

4 – REPM
Actions are classified according to REPM Levels. The levels spans from 1 to 5 (See section 2.2).

5 – Relation
The relation denotes a dependency of sorts between Actions. This generally means that an Action on e.g. REPM Level 3 is dependent on that one or more Actions at a lower level are fulfilled, note that this is not always the case, e.g. one of two dependent Actions can be optional (See below).

2.1.1.1 Optional Actions
In some cases Actions in the REPM Model are optional, i.e. they do not have to be satisfied by a Requirements Engineering process (Figure 2.4). There are two types of Optional Actions; Optional (denoted “Opt” in the model) and Optional Groups (denoted by “OG” in the model). *Opt* stands for optional and this Action is voluntary, i.e. it does not have to be satisfied in order for a Requirements Engineering Process to reach a certain REPM Level (See section 2.2). *OG* denotes a group of Actions of which at least one has to be satisfied in order for a project’s Requirements Engineering process to qualify for a certain REPM Level. An example of this can be seen in Figure 2.4. In this example, we have three Actions; Risk Assessment – individual, sets and selected. Each of these has an additional group identifier, i.e. OG1.01, OG1.02 and OG1.03. This denotes that all three belong to the same Optional Group (OG1) and the last number denotes the group id of each Action. For a project to reside in REPM level 3 OG1.03 has to be satisfied. If a project is to satisfy REPM level 4 at least one of OG1.01 and OG1.02 has to be satisfied, i.e. if OG1.03 is satisfied or not does not matter if REPM level 4 is the goal.
2.2 Requirements Engineering Process Maturity

Every Action is mapped to a certain Requirements Engineering Process Maturity (REPM) Level spanning from 1 to 5 (See below). The motivation behind placing an Action in a certain REPM Level is comprised of two main parts, cost and complexity. Cost denotes how much resources, e.g. man-hours and/or money, must be spent in order to satisfy an Action – the more costly the higher the level. An example can be the Action of E.1.a1 Ask Executive Stakeholders which is situated on REPM Level 1 and E.1.a2 Research Stakeholders which is situated on level 2. It is cheaper in terms of man-hours to ask the executive stakeholders, i.e. the ones ordering the system, than investigating who the stakeholders are. Complexity denotes how complex a certain Action is. An example can be the Action of E.3.a3 Technical Domain Consideration (taking the system’s technical operating environment into consideration) which is situated on level 1 and E.3.a1 Human Domain Consideration (taking organizational and political factors into consideration) which is situated on level 4. For a software development company it is often a necessity to investigate and take the technical aspects of the system’s operating environment into consideration, but how many take political factors, e.g. an employee who is reluctant to speak freely when the boss is present, into consideration when eliciting requirements?

In essence the REPM levels denote how advanced and mature a Requirements Engineering process for a certain project is, i.e. a higher level denotes a higher level of maturity. This is however not the same as saying that all companies can, or even should, try to get all their projects to reside on the highest level. It costs resources to reach a higher level of maturity, whether it is the Requirements Engineering process or any other part of the development process. A REPM level of three may be adequate for a certain
Chapter Two – The REPM Model

type of project. What level a certain type of project should reside upon we leave to the organization handling the project.

The five REPM Levels are presented below.

2.2.1 REPM 1 – Initial (Wood)

The Actions at REPM Level 1 are what is needed to make a basic requirement specification. The Requirements Engineering process is very slim at this level and not necessarily repeatable, i.e. not something that is repeated in the same way project after project, but rather a thing of chance.

Organizations typically do not provide a stable environment for development. During a crisis, projects typically abandon planned procedures and revert to coding and testing. In addition no validation or review of the requirements takes place.

2.2.2 REPM 2 – Basic (Bronze)

Level 2 denotes a more structured and complete Requirements Engineering process than level 1. The goals for this level are:

A. Introduction of traceability
B. Introduction of validation of requirements
C. Introduction of a standardized structure for the documentation produced as a result of the Requirements Engineering process, i.e. the Requirements Document
D. Stakeholder identification

An organization at this level has introduced policies that ensure that requirements are specified and documented in a standardized way. This ensures repeatability inside a project, i.e. every requirement is elicited, documented and verified in the same way. The Bronze level in general denotes that an organization has devoted resources to the Requirements Engineering process as a separate entity in the Software Engineering process as a whole.

At this level the environment of the system being produced is only studied in passing, i.e. no notice is taken to the application domain or the business processes already present. Stakeholders are identified. Requirements may be insufficient in detail and/or in number, as well as incompatible with the system’s operating environment.

We expect to find that Companies/organizations producing projects at this level tend to be smaller, up to 50 co-workers, and tend to be fairly young, i.e. not been in operation for more than 5 years. If we look at the domain of the organizations at this level we will probably find smaller companies involved in the development of information systems.
2.2.3 REPM 3 – Formulated (Silver)

Level 3 denotes a more active examination of the system environment. The goals for this level are:

A. Application domain and processes are studied and taken into consideration
B. All stakeholders are consulted
C. Dependencies, interactions and conflicts between requirements are taken into consideration
D. Requirement categorization and prioritization
E. Requirements re-prioritization
F. Peer-reviews
G. Risk assessment

The system’s environment is studied in greater detail, not only the technical aspects but also the demands coming from the application domain, as well as the business processes which the system should support. This ensures a deeper understanding of the environment in which the system is to operate, and subsequently enhances the ability to design the requirements. All stakeholder groups are consulted and reviews are conducted, i.e. peer-reviews that may or may not be contractually bound. The requirements are prioritized and re-prioritized in case of new releases and/or new requirements. Furthermore interactions between the requirements are mapped through the use of e.g. Interaction Matrices. This alleviates the risk of conflicts. In addition to this risk assessment is conducted on selected requirements, i.e. requirements that are likely to change.

At this level no structured risk assessment is performed on sets or individual requirements. Furthermore no consideration is taken to the human domain, e.g. political and emotional factors. In addition the question of whether or not the system will add value to the organization is not raised.

We expect to find that Companies/organizations producing projects at this level tend to be moderate in size, up to 100 co-workers, and tend to be fairly seasoned, i.e. been in operation for 6 to ten years. If we look at the domain of the organizations at this level we will probably find established companies involved in the development of information systems.

2.2.4 REPM 4 – Developed (Gold)

Level 4 denotes a more active and mandatory examination of risks and the true value of the system seen from the organization where the system is to be implemented. The goals for this level are:

A. Human domain consideration
B. Business domain consideration
C. Advanced risk assessment
Chapter Two – The REPM Model

D. Advanced traceability

Consideration is taken to human domain aspects, e.g. political and emotional factors which can greatly influence requirements sources. How the system developed makes a contribution to the business at hand is also studied. This will help to drive the elicitation process and goals are set higher, i.e. more can be expected from the system on completion. The requirements are refined through the use of scenarios at a larger scale than before, and the requirements are validated through inspections (walkthroughs). Risk assessment of the requirements (individual and/or sets) is done, e.g. risk is weighed against potential gain of a certain requirement. Traceability options are more advanced and cover documents preceding the requirements document, e.g. links pre-study documents to the relevant requirements.

This level is fairly advanced and all the major components of a well scrutinized and standardized Requirements Engineering process are present. However a planned and systematical requirements reuse structure is not present. The system architecture is not studied and may bring unwanted results in terms of unexpected interactions between subsystems.

We expect to find that Companies/organizations producing projects at this level tend to be large in size, up to 300 co-workers and above, and tend to be fairly seasoned, i.e. been in operation for more than ten years. If we look at the domain of the organizations at this level we will probably find established companies involved in the development of information systems as well as companies involved in critical system development.

2.2.5 REPM 5 – Advanced (Platinum)

Level 5 denotes advancement from the previous level when it comes to reuse and architectural considerations. The goals for this level are;

A. Requirements reuse
B. Rejected requirements documentation
C. Architectural modeling
D. Advanced validation
E. Advanced requirements re-prioritization

Requirements reuse is taken into consideration and reuse is done when possible. Also rejected requirements are documented as a part of the Requirements Engineering process. This documentation offers clarity (you have what should not be implemented on paper) as well as material for future reference. System model paraphrasing is used to further validate requirements as well as the creation of architectural models to map the communication between the system as a whole and the environment, but also the communication within the system, e.g. between sub-systems. The requirements are re-prioritized with regularity, independent of whether of not something happens.

We expect to find that Companies/organizations producing projects at this level tend to be large in size, up to 300 co-workers and above, and tend to be fairly seasoned, i.e. been
in operation for more than ten years. If we look at the domain of the organizations at this level we will probably find established companies involved in the development of information systems as well as companies involved in critical system development with very large projects demanding vast resources.

### 2.2.6 Adding to the model

Adding to the model demands the following:

- Decide if one wishes to add an Action or a SPA. If an Action is to be added it is very important that the Action is added under the right MPA or SPA.
- Make sure that an already present MPA, SPA or Action is not the same or similar to the one being added.
- Make sure that the description of the added item is correct and adequate.
- Decide on what REPM Level the added Action should preside.
- Evaluate if the Action is to be linked (have a relation) with any other MPA, SPA or Action.
- Make sure that the added item gets the correct name identifier.

In addition it is possible to convert an existing Action to a SPA if one feels that an Action is too general or needs to be split up into several Actions.

If an Action is converted to a SPA the same rules as mentioned above apply, in addition it is important to notice that a SPA can not be the finite level – Actions have to be present at the bottom of the hierarchy, i.e. a SPA has to have Actions or other SPAs under it.

We do not recommend adding items lower than 4 levels. An item added at even lower levels may unnecessarily complicate the model.

### 2.3 Usage of the REPM model

The REPM model can be used for several different things. Primarily it is intended for evaluation of the maturity of the Requirements Engineering process of a certain project. The reason for it being designed towards projects and not the whole of an organization like e.g. CMM is the fact that the Requirements Engineering process varies with projects. This variation can be a result of an incomplete Requirements Engineering process, i.e. a non-repeatable one (equivalent a process at REPM level 1, See section 2.2.1). But it can also be a result of the fact that different projects merit different levels of Requirements Engineering. An example of this can be an in-house project where for example risk assessment does not have to be done to the same extent as in a traditional developer-customer relationship, as the risks have already been assessed before the go-ahead was given. The usability of the REPM model does not decrease, according to our opinion, due to its project focus however. If one desires to measure the REPM level of the organization this is still possible through measuring all of the projects – thereby evaluating everything produced by the organization.
A direct spin-off effect of using the REPM model as an evaluation tool is that it produces a blueprint of the status of the Requirement Engineering process of a certain project. This can be used to see what is done, what is not, and in continuation what can be done to improve the process. With improvement we do not necessarily mean making it to the next REPM level, but rather getting to the level best suited for the project at hand. It is important to notice that a REPM level of five is not necessarily the “best” thing. All process improvement cost resources and one has to weigh the benefits of a potential improvement against the cost of achieving that improvement. Whether or not a change is to be made, or on what REPM level a certain project should reside on for that matter, is up to the people in charge. Our goal is only to give developers an easy way to assess the situation, what they do with the data is up to them.

In addition to using the REPM model for evaluation purposes it can be used as a framework of what things to do during a project. The model is not designed for this, e.g. no chronological ordering of the Actions is made, but it is conceivable that a summation of the Actions (See Appendix I - Action summary) can act as a checklist of sorts.

### 2.3.1 Project Evaluation

In Appendix III we provide a checklist to use when evaluating projects. The checklist questions follow the model and primarily its Actions. The default procedure is that one question mirrors one Action in the model. However, in certain instances more than one question is posed to ascertain if a certain Action is completed.

<table>
<thead>
<tr>
<th>E. Requirements Elicitation</th>
<th>Action UID</th>
<th>YES</th>
<th>NO</th>
<th>Comment if NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Do you reuse requirements from other systems developed in the same application area?</td>
<td>E.a1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**E.1 Stakeholder identification**

2. When determining whom the stakeholders are for a system, do you ask the people ordering the system, whom they think are the stakeholders?  

3. Do you do you own research determining who the stakeholders are?  

**Figure 2.5** Extract from REPM Evaluation Checklist.

The basic structure of the REPM evaluation checklist is the same as for the model. It is sorted according to MPA and SPA. The *Action UID* denotes which Action in the REPM model the question is linked to. This so that the person(s) answering the question can look in the model for clarification if need be. In some instances a general question is asked with a MPA or SPA as a template, this to determine if any of a group of Actions is satisfied, i.e. a question like “do you do any risk assessment” is based on the SPA preceding Actions about particular risk areas. If the answer is “no” on the first question...
Chapter Two – The REPM Model

there is no need to answer questions on specific risk assessment issues. Furthermore if “no” is checked the person answering the question can (and should) clarify why, i.e. to make it possible to deem if an Action is Satisfied-Explained (See section 2.3.2).

2.3.2 The Result

When the REPM evaluation checklist is completed the results are added up according to REPM level. A project has to complete all Actions at a certain REPM level to qualify for the level in question (special rules for optional Actions, See section 2.1.1 and Actions deemed Satisfied-Explained), i.e. for a project to reside on a REPM level of 3 all Actions at level 1, 2 and 3 have to be completed. A “perfect” example of this is:

\[
\text{REPM level of Requirements Elicitation} = 3 \\
\text{REPM level of Analysis and Negotiation} = 3 \\
\text{REPM level of Requirements Management} = 3 \\
\Rightarrow (3+3+3)/3 = 3
\]

This gives a total REPM level of 3. If there are some completed Actions on level 4 and 5 these are disregarded in the overall REPM level (they are however present in the presentation of a project evaluation, see below).

A more realistic example we think would be when the REPM level of different MPAs differ, e.g.

\[
\text{REPM level of Requirements Elicitation} = 3 \\
\text{REPM level of Analysis and Negotiation} = 2 \\
\text{REPM level of Requirements Management} = 4 \\
\Rightarrow (3+2+4)/3 = 3
\]

This also gives a REPM level of 3 in total, but the difference is that it is easy to see that the MPA of Analysis and Negotiation is at a lower level than the other two. At this stage it is possible for the organization testing its project to see in what area the process may be insufficient.

It is important to notice that the REPM levels should be homogenous, ensuring that dependencies between the different MPAs are satisfied and that there is a consistent level of maturity in the work being done, i.e. a chain is only as strong as its weakest link. An example of this can be an organization with REPM level 1 for the MPA of Requirements Elicitation and a REPM level of 5 when it comes to the MPA of Requirements Management. This would mean that advanced requirements validation is performed on requirements that are poorly elicited in the first place.

2.3.2.1 Satisfied-Explained Actions

Another factor that has to be taken into consideration when using the REPM model for project evaluation is that certain projects do not merit the use of all Actions. This can be due to many reasons, e.g. internal projects (in-house development) may not merit the use
Chapter Two – The REPM Model

of e.g. a detailed risk analysis or the need to research stakeholders etc. A company devoted to carrying out projects in special environments, i.e. a variant of the traditional customer-developer environment, may deem certain Actions unnecessary. An example can be a company where the developer and the customer both have specialized on a certain domain and “speak the same language”. The need for extended clarification and validation of requirements may not be needed, e.g. the construction of prototypes can be omitted. This is not the same as optional Actions (See section 2.1.1.1) however. Optional Actions are Actions that we consider not to be vital to the Requirements Engineering process, but may be a good complement. All Actions not marked optional are considered vital.

In order to take all of these factors into consideration we have chosen to add a new term to the REPM model, *Satisfied-Explained*. This expression denotes an Action that is not completed - but the organization doing the evaluation deems the Action not applicable to their project. Satisfied-Explained is used to counter the effect of misleading results when evaluating a project.

An example of this can be a company not using the Action of *E.4.a3 In-house Scenario Creation*. Let us say that the developing company works in, and is specialized on, one domain. All of the customers are also specialized in the same domain. In addition the developing company sells only one system, highly specialized to suit the domain and it is only adapted to some extent for every customer. The developing company deems the Action *E.4.a3 In-house Scenario Creation* to be unnecessary due to the fact that they have much experience of adapting and selling the system, and do not need to construct in-house scenarios to clarify and validate the requirements at hand. When doing an evaluation of the project in question the REPM level would not even reach level one (REPM 1) due to the fact that *E.4.a3 In-house Scenario Creation* (which is on level 1) was not completed. The rest of the Actions up to REPM level 5 could be fully completed but the REPM level of the project would be zero anyway. This is misleading and does not give an accurate picture of the REPM level of the project. For this reason Actions are considered satisfied both when actually completed and when they are deemed to be inapplicable and/or irrelevant to the project at hand, in this case the Action is said to be Satisfied-Explained (not completed).

The organization doing the evaluation makes the distinction when an Action is to be considered Satisfied-Explained. Important to notice is that an Action is not deemed Satisfied-Explained for reasons like lack of time, lack of money, lack of know-how or just “did not think of it”. The lack of time and money is fairly self explanatory, as is “did not think of it”. Know-how however refers to an instance where the competence to implement an Action is not present in the organization. Know-how can also refer to the fact that a certain Action’s benefit (the positive effect of conducting a certain task) is not obvious. It is easy to dismiss Actions (put them into the category of Satisfied-Explained) and this is a substantial risk. However this is up to the person(s) evaluating the project.
Chapter Two – The REPM Model

The use of Satisfied-Explained Actions, i.e. if an organization deems that one or more Actions in the REPM model falls into this category, is subjective. The one(s) doing the project evaluation make the distinction. An Action can thus after an evaluation have three states:

1. **Completed** - The Actions is done.

2. **Not Completed** - The Action is not done for any number of reasons.

3. **Satisfied-Explained** - The Action is not completed or completed partially, but the main thing is that the Action as formulated in the REPM model is not applicable to the Requirement Engineering process of the organization evaluating the project.

### 2.3.2.2 Result Presentation through Diagrams

![Figure 2.6 Example of Total/Completed/Satisfied-Explained Actions.](image)

The use of writing out the REPM level of a certain project in numbers is a fairly crude way of measuring the Requirements Engineering process. It can be a good overall indicator but there are better ways of presenting the results of a project evaluation using the REPM model. In Figure 2.6 an example of this is given. There are three things presented; the total number of Actions at each REPM level, completed Actions and Actions that fall under the category of Satisfied-Explained. In this example REPM level 1 is not achieved (only 8 out of 10 Actions are completed). If we look at the Satisfied-
Chapter Two – The REPM Model

Explained graph line the two remaining (not completed) Actions are Satisfied-Explained – thus a total of 10 Actions are satisfied rendering the project on REPM level 1. REPM level 2 is not achieved, not even when taking the Satisfied-Explained Actions into consideration. In the graph the line denoting the Satisfied-Explained Actions is special. It really denotes completed Actions plus the Satisfied-Explained Actions. This is to enhance the readability and the usability of the graph. A summary of the Actions completed and Satisfied-Explained is the real measurement of a companies Requirements Engineering process maturity level. It is easy to differentiate between the two (completed and Satisfied-Explained) however as the line denoting the completed Actions is also present.

For the organization doing the evaluation the area between Total Actions-line and the Satisfied-Explained-line is the most interesting. It represents the space for possible improvement of the Requirements Engineering process. The area between the Completed-line and the Satisfied-Explained-line represents the level of inapplicability of the model to a certain projects Requirements Engineering process (called model lag in the future).
2.3.3 Result Presentation – an Example

Below we give an example of how a project evaluation can be presented using the fictional project MSI. No detailed interpretations of improvement possibilities or the consequences of these are offered here. Examples of this can be found in Chapter 4 – where the actual project evaluations are presented. The goal here is only to give an introduction to the presentation design, i.e. the graphs and what they represent, to prepare for the real project evaluations being presented later.

Project MSI has the following statistics:

2.3.3.1 Action Summary

Total number of Actions on each REPM level compared with the amount of Actions completed and Satisfied-Explained in project MSI.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that the MSI project resides on REPM level 1, i.e. Wood, and that 8 out of 10 Actions are completed. In addition we can see how far the MSI
Chapter Two – The REPM Model

The project has to go in order to advance to REPM level 2 (Bronze), 3 (Silver), 4 (Gold) and 5 (Platinum) regardless of MPA. This is an overall summary of the Requirements Engineering process of the MSI project. Interesting to notice is that the model lag is fairly large at REPM level 2 and 3.

In addition one can see that the MSI project resides on REPM level 1 but has a short distance to REPM level 2.

2.3.3.2 MPA of Requirements Elicitation

Total number of Elicitation Actions on each REPM level compared with the amount of Elicitation Actions completed and Satisfied-Explained in project MSI.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 2.8 MSI project chart 2.
Chapter Two – The REPM Model

From this chart we can ascertain that the MSI project resides on Silver when it comes to the MPA of Elicitation. Furthermore it is interesting to notice that the model lag is fairly low within this MPA.

To evolve the Requirements Engineering process to Platinum only one Action has to be satisfied additionally (on REPM level 4).

2.3.3.3 MPA of Requirements Analysis and Negotiation

Total number of Analysis and Negotiation Actions on each REPM level compared with the amount of Analysis and Negotiation Actions completed and Satisfied-Explained in project MSI.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that the MSI project resides on Bronze and is only 1 Action from Silver. The model lag on REPM level 3 is more substantial than in the MPA of Elicitation.

Figure 2.9 MSI project chart 3.
Chapter Two – The REPM Model

Important to notice here and generally is that Satisfied-Explained Actions should not be considered as a convenient excuse not to complete Actions, but it should rather be used only when necessary. In Figure 2.9 the diagram denotes that no Action is completed at all on REPM level 1 and 2, and that totally 6 Actions are accredited to the category of Satisfied-Explained. This is a fairly high number and it can be an idea to scrutinize the reasons for that many Actions to fall under said category.

2.3.3.4 MPA of Requirements Management

Total number of Management Actions on each REPM level compared with the amount of Management Actions completed and Satisfied-Explained in project MSI.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that the MSI project resides on Wood (completed). An REPM level of 2 is fairly close to be completed; only 1 Action is needed additionally. The model lag is greatest on REPM level 2, but overall it seems manageable.

In conclusion one can ascertain that the MPA of Management is the largest problem area in the MSI project, i.e. 1 Action to achieve REPM level 2, and 3 Actions to achieve...
Chapter Two – The REPM Model

REPM level 3. Looking at Figure 2.8 and 2.9 one can see that both MPAs reside on REPM level 2, and only 1 Action in MPA of Analysis and Negotiation is needed to achieve REPM level 3.

In the presentation above a completed project is presented. It can be beneficial to evaluate a project that is ongoing – looking at the Requirements Engineering process over time until project completion. This can give information about many things such as when Actions are completed, time spent completing them and so on.

In addition to presenting project evaluations in the summary form of diagrams it is also possible, and in some instances beneficial, to present the raw data, i.e. the Project Evaluation Checklist in order to see details.

2.4 Conclusion

The design of the REPM model is centered on changeability, i.e. the notion that the model will grow in size and complexity in the future. Using the model as a general process evaluation tool the model should be kept rather general, i.e. high-level Actions, leaving much to interpretation and thus making the model “fit” the reality of projects in industry. On the other hand it is possible to tailor the model to fit a certain organization or projects type. This would allow for low-level Actions.

The term model lag denotes the inapplicability of the model to projects in industry. The idea is to not penalize organizations with a Requirement Engineering process that is good for their particular situation. A certain project’s Requirements Engineering process may be adequate in spite of the fact that certain Actions are not performed. The Actions falling under this category are said to be Satisfied-Explained. Important to realize here is that the placement of a certain Action in this category should only be done after careful consideration. The Satisfied-Explained category is not to be considered as an easy way out by the one making the decision, i.e. the project responsible, but rather possibility to use the model in spite of the fact that a Requirements Engineering process may be special in some way and thus being at odds with the general model.

Presentation of a project evaluation can be done in any number of ways. One important thing to notice is that diagrams in combination with text can be a powerful technique. The project evaluation example presented above is a fairly simple but, we feel, effective way of showing the results and conveying the information to the reader. We have chosen to use this form of presentation throughout the paper (See section 4).

In Chapter 4 several projects from industry are evaluated. The diagrams used above are presented as well as an analysis of them, how the Requirements Engineering process can be improved, and what this improvement would mean for the project’s Requirement Engineering process. The test case shown above aims to familiarize the reader with the presentation technique we have chosen to use. It is important to notice that the presentation could be modified and/or extended if necessary as long as the results presented are accurate.
Chapter Three – REPM Model Validation

3 Chapter Three – REPM Model Validation

The purpose of this chapter is to present the first round of validation performed to validate and improve the REPM model. This was done by interviewing a senior project manager/team leader at SchlumbergerSema in Karlskrona/Sweden. The results of the interview are presented below along with information on what steps were taken as a response to the data and feedback gathered during the interview.

This chapter is divided into three main parts. First a description of how the validation was conducted and structured, how the interview questions and the interview were designed and how the interview subject was chosen according to certain criteria. Second the result of the interview is presented in summary along with suggestions by the interview subject on how the model can be enhanced. In combination with these improvement suggestions our own notes pertaining to what was done are presented. Last a summary where we draw conclusions of the REPM model validation.

3.1 Validation Method

To be able to elicit the information needed the use of structured interviews was applied [22]. This gave the researchers structure and also the freedom to promote a dialogue with the interview subject during the interview without straying too far from the subject at hand. By conducting the interview in person the risk of misunderstanding the questions was reduced. On the other hand the researcher might have influenced the answers by his presence and/or actions [22] [25]. It is also important to notice that the researcher can influence the interview subject in a good way, i.e. a good attitude can make the subject more loquacious.

In general one could say that the interview was qualitative in nature, e.g. a secondary objective, or rather a positive spin-off effect of the validation process, was that the interviewee was used as a sounding board. Important to notice with a qualitative approach is that there are many possible interpretations of the problem at hand, and thus the result gathered from the interview [21]. Furthermore we realized that the basic task of getting answers to the questions we wanted answered was not always easy, it is difficult to steer interview subjects [23]. The key to alleviate this problem was preparation in terms of structuring the interview before hand.

3.1.1 Selecting Interview Subjects

The interview subject was chosen with several criteria in mind. First the subject had to work in a company/department that is involved with Software Engineering/Development towards customers outside their own organization. This to have experience of a more typical developer – customer relationship. Second the subject had to be in a Senior position within the development group/team (more than 10 years experience), e.g. project managers or ideally senior developers in charge of the RE process. Third the company
employing the subject should ideally have long experience in the field of Software Engineering, i.e. at least 10 years.

For time reasons the selection was done using convenience sampling, i.e. we talked to companies that we had a previous relationship to. A random selection from all of the companies that fit the profile described above would unfortunately have taken to much time and demanded resources that we did not have. The negative aspect of convenience sampling is that the selection method is not statistically satisfactory [25] [26].

3.1.2 Interview Questions

The questions posed during the interview are designed in such a way that they in themselves give as little foundation for error and misinterpretation as possible, e.g. questions can be leading, give rise to prestige bias and so on [22] [25]. To avoid this the questions have been studied not only by the two researchers but also Mikael Svahnberg (Doctoral Student, Advisor) and Kennet Henningsson (Doctoral Student).

The interview questions are divided into three parts; an introductory part, a sort of starting point to establish certain general issues, a detailed part consisting of questions that concerned more specific details of the model, and a final part consisting of questions concerning what may be lacking from the model, or if the reader/interviewee feels that something should be added to the model. The idea of warm-up and cool-off periods is applied, thus concentrating the vital questions in-between [22]. The questions can be viewed in Appendix II.

3.1.3 Presenting the Interview subject

The person we chose to interview was Lars Göran Cronberg at SchlumbergerSema. Mr. Cronberg is more than qualified both according to the criteria posed and our own opinion. The company SchlumbergerSema also meet the criteria posed.

A copy of the REPM Model (Version 0.93 - This version can be obtained from the authors) was sent to the interview subject beforehand so that he could study the model. The interview subject was asked to study the model before the interview, and note the time he spent studying/reviewing the model.

The interview was conducted on-site at the location of the interview subject’s work place, this not to inconvenience the subject more than necessary. The interview was recorded with digital camcorder and later transcribed, not word-for-word but rather in summarized answers to the questions posed (a full transcription can be obtained from the authors). Notes were also taken during the interview assisting the dialogue in terms of drawing diagrams and making notes in cooperation with the subject.

At the conclusion of the interview the subject asked for more time to review the central questions (question 24-66 See Appendix II) in order to be able to give them more thought. The subject complemented his answers to the questions in written form within four days time.
Chapter Three – REPM Model Validation

Directly after the interview a first round of summarization of the interview result was done in order to extract as much data as possible from the interview. As the secondary (complementary) answers arrived in written form the summary was also complemented/added to. To clarify and validate the answers from the complementary part of the interview a secondary interview was conducted over telephone.

After the interview was concluded, and the final summary produced, the interview subject was offered to view the result before the material was published and/or acted upon. The interview subject consented to the material presented in this chapter.

3.2 Interview Result

Lars Göran Cronberg is a senior consultant/project manager at SchlumbergerSema and handles projects varying in size, e.g. from hundreds of man-hours to tens of thousands. Mr. Cronberg considers the area of Requirements Engineering to be very important and crucial part in his projects, and in software engineering in general. About half of Mr. Cronberg’s time is devoted to the area of Requirements Engineering.

Studying the REPM model Mr. Cronberg felt that it was easily understood. The structure was clear and the division - from MPAs to SPAs and Actions was logical and uncomplicated (MPA = Main Process Area, SPA=Sub-process Area).

Mr. Cronberg felt that the grading of the Actions (REPM level 1-5) was good, but he did not understand clearly how the REPM levels were created, i.e. what prompted an Action to be on a certain REPM level. The scale as such and the ability to grade projects according to it – making it possible to evaluate and get information about improvement possibilities was something Mr. Cronberg saw as very beneficial. The results of a project evaluation according to the REPM model and the presentation of them was also discussed. The use of diagrams and an approach where one could compare results from evaluations done in different project stages was discussed.

In summation Mr. Cronberg felt that the REPM model was a good initiative and could be helpful in improving the Requirements Engineering process. Mr. Cronberg could very easily see himself use the model for project evaluation and as a checklist of sorts during projects.

3.2.1 Suggestions for Improvement

Mr. Cronberg also suggested a number of improvements which we present below. After every suggested point of improvement there is a summation of what action was taken as a response. Important to notice is that every suggestion was considered carefully – even thought all suggestions were not acted upon by changing the model.
1. The Actions in the model should be sorted in an order following their REPM level.

The Actions are sorted by the unique identifier, e.g. “E.3.a1”, and not the REPM level. The sorting by REPM level would be ineffective due to the fact that the model is expandable and prone to change. If Actions are sorted by REPM level and named (given a unique identifier by the same order) it would result in a need to update the entire model every time an Action is added/deleted.

An example of this could be: E.3.a1 which has REPM level 4, E.3.a2 has REPM level 3, E.3.a3 has REPM level 1, E.3.a4 has REPM level 4 and E.3.a5 has REPM 3. If the Actions were sorted by REPM level it would result in the order being E.3.a3, E.3.a2, E.3.a5, E.3.a1 and E.3.a4 which can be confusing. If we were to rename the Actions to suit the sorting we would have to do so every time an Action was added/deleted. For these reasons we choose not to implement this suggested modification to the REPM model.

<table>
<thead>
<tr>
<th>UID</th>
<th>REPM level</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.3.a1</td>
<td>4</td>
</tr>
<tr>
<td>E.3.a2</td>
<td>3</td>
</tr>
<tr>
<td>E.3.a3</td>
<td>1</td>
</tr>
<tr>
<td>E.3.a4</td>
<td>4</td>
</tr>
<tr>
<td>E.3.a5</td>
<td>3</td>
</tr>
</tbody>
</table>

Sorting by REPM level

<table>
<thead>
<tr>
<th>UID</th>
<th>REPM level</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.3.a3</td>
<td>1</td>
</tr>
<tr>
<td>E.3.a2</td>
<td>3</td>
</tr>
<tr>
<td>E.3.a5</td>
<td>3</td>
</tr>
<tr>
<td>E.3.a1</td>
<td>4</td>
</tr>
<tr>
<td>E.3.a4</td>
<td>4</td>
</tr>
</tbody>
</table>

2. Add an additional SPA under the area of Requirements Elicitation named “Project Domain”. This should include facts like project size, type of project, type of organization, work models and tools.

The addition of a new SPA under the MPA of Requirements Elicitation that depicted project size, type of project, type of organization, work models and tools was not implemented. We acknowledge that customers may have requirements to that fact, e.g. how the project should be managed, what tools should be used, resources committed and so on. The reason for not changing the REPM model to reflect this is that we deem this not to be a part of the Requirements Engineering process. We felt that this rather is a part of the product development planning and/or the project contract management process.

3. The two SPAs Stakeholder Identification and Stakeholder Consulting should be merged into one SPA. They are conducted in combination for the most part anyway.

We acknowledge that the SPAs of Stakeholder Identification and Stakeholder Consulting are linked, and ideally done after each other. In our opinion however they are not to be merged into one SPA due to the fact that they are independent. In projects there are cases when stakeholders are identified and not consulted, i.e. identification does not necessarily mean consulting. We did however decide to move
Chapter Three – REPM Model Validation

the SPA of Stakeholder Consulting from E.3 to E.2 in order to group the two together due to their obvious relation pointed out by the interview subject.

4. Add a SPA under the MPA of Analysis and Negotiation called “Contract Management”. This Action should handle all the information about the contract drawn up by the developers and the customers.

Contract management is not a part of the Requirements Engineering process, but rather the project management/planning area. Due to this reason we deemed that this area should not be part of the REPM model.

5. The SPA of System Boundaries Definition is covered inadequately.

During additional conversation with the interviewee it became clear that the Action A.1.a1 Boundary definition through categorization under the SPA of A.1 System Boundaries Definition was unclear in its definition (the explanation under the Action was inadequate) and the explanation needed to be extended. This was done.

6. The SPA of Requirements Prioritization should be extended with Actions dealing with the change in priority over time.

A SPA was added under A.2 Requirements Prioritization with four Actions. These describe that requirements’ priority may change over time, due to events and/or changes in the environment. This was a new addition to the REPM model.

7. The Actions under the SPA of Requirements Risks should be divided with different levels of REPM. Certain things of risk management are according to the interview subject present already at earlier levels.

We agreed that some risk management should be present at an earlier REPM level and the Action A.3.a3 Risk Assessment – selected was put on REPM level 3 instead of 4. This change also affected the Action M.4.a1 Volatile Requirements Identification which was also put on REPM level 3 (from 4), i.e. you have to identify the requirements that are prone to change in order to assess the risk of them changing.

8. Under the MPA of Requirements Management a SPA with the title “System Documentation” should be added. This to handle the information of which kind of documents should be written during the development process, i.e. what document deliverables there should be.

System documentation is an important part of the system deliverables and should be specified as part of the requirements document. We added a SPA called M.5 Documentation Deliverables. Under this SPA three Actions were added dividing the documentation into three main groups; user-, system- and management documentation. Each of these groups is fairly large, i.e. cover a large field of practice and additional Actions may be added in future releases of the REPM model.
Chapter Three – REPM Model Validation

9. There should be Actions or information added that handles the involvement of third-party products and developers.

We consider that the Action E.3.a3 Technical Domain Consideration covers this area. An addition to the description of this Action was made to clarify this.

10. In the MPA of Requirements Management something should be added to point out the usefulness of going from the documents produced during development to see what parts needs to be described in the Requirements Document, i.e. using the documentation to validate the requirements document.

Using produced documentation to validate the requirements is a very good and useful approach for a seasoned developer/project manager. The problem is that you have to make a distinction – should the documentation govern what requirements are present or should the requirements govern what documentation is present? This was a distinction that we preferred not to incorporate into the model at this stage. This does not however diminish the usability of this approach, e.g. as a form of validation.

11. All the Actions under the SPA Domain Knowledge should be turned into SPAs themselves and have detailed Actions under them explaining what should be done. This due to the fact that much time and money are spent on this area. The depth of other areas in the model, e.g. under the MPA of Requirements Management is covered in greater detail.

This observation, and subsequent suggestion to further granulate the area, was very astute and we agree fully. This area needs to be developed further due to its importance to the Requirements Engineering process and the system development process in general. This is especially important to notice in the case of longer and more resource consuming projects where the system domain becomes more apparent, and the development team learns how to ask the “right” questions over time, i.e. the more you know about the domain the more detailed are the questions about the domain.

However at this stage of the REPM model’s development the addition/modification commands resources that are not available. Important to notice is that we consider this an important “next step” in the REPM model’s development.

12. The model should be adapted to suit both Internal and External projects. Some Actions are not valid and justified in both areas and this should be pointed out. E.g. risk management may be superfluous in internal development projects due to the fact that in many cases the risks are assessed beforehand by the internal customers.

This point was well taken and is considered for future versions of the REPM model. This area was not incorporated in the REPM model due to the fact that the model was primarily designed for evaluating external projects (developer-customer relationship). In addition one could argue that an internal project might be suited to reside on an REPM level of lower rank. This enables the user of the REPM model to avoid certain
Chapter Three – REPM Model Validation

Actions, e.g. associated with risk assessment. In addition the use of Satisfied-Explained Actions (See section 2.3.2.1) helps with this problem, i.e. all Actions do not have to be completed if they are considered inapplicable for a certain project.

13. An “Anti-Requirements list” should be made explaining what the requirements are and especially what they are not, i.e. what things are not covered by a requirement specified.

This point was well taken and an Action was added - A.a4 Ambiguous Requirements refinement to reflect this. It was added as an optional Action due to the fact that it may be a complement if the specifier of the requirement feels that a certain requirement is equivocal, and suspects that the stakeholders read more into a requirement than is really present.

14. It is important to root the importance of Requirements Engineering with the customer at an early stage – basically sell the idea of save-money-through-a-good-requirements-engineering-process. This should be present in the model.

We fully agree with this point and sadly a requirements engineer has to sell the idea of Requirements Engineering to the customer in order to gain resources needed to carry out the work in question. However we do not feel that this is a part of the Requirements Engineering process as such, but rather the sales process. We are fully aware that in “real” industry projects the distinction may be somewhat in the “gray area”, i.e. the whole system development process is a give-and-take situation where certain things (ideas) have to be clarified and basically sold to the customer.

15. The presentation of the result of a project evaluation can be better if diagrams are used.

The use of diagrams for presentation was considered from the early stages of the REPM model’s development. However Mr. Cronberg’s suggestions gave us some new ideas and validated the idea of using diagrams in the presentation (See section 2.3.3).

3.3 Conclusion

Designing, preparing and conducting interviews requires meticulous preparation and planning. In addition, the results gathered can often be interpreted in several ways and can be prone to some misunderstandings. A follow-up session can be a good way to clarify and further help articulate questions. The main goal of the validation presented in this chapter was to get ideas for how to improve the REPM model. The follow-up session gave us further clarifications so that nothing of the information gathered was misunderstood and/or dismissed without careful consideration. We feel that our skills in this regard have improved and we learned many important lessons during the day-to-day work.
Chapter Three – REPM Model Validation

During the interview we got many good suggestions about how the model could be improved and we have taken all points under careful consideration. The help and feedback offered by Mr. Cronberg is deeply appreciated.

A further granulation of the model was in many cases needed to meet the suggestions offered by Mr. Cronberg. In some cases we opted not to make changes, e.g. in the case of domain knowledge. The main reason being that we did not have the resources, i.e. time, to make the extensive changes necessary. However there are a couple of additional reasons for watching out when adding to the size, and in some cases, complexity of the REPM model. Adding to the size of the model may in turn have the effect that the process of evaluating a project takes more time and subsequently costs more. However this is secondary to the problem of complexity. Many people may find a large and complex model intimidating, i.e. hard and time consuming to learn and use. This would probably not be the case as far as adding to the specific area of domain knowledge as is the case above, but it may become a problem in the future when granulation has progressed.

Furthermore it is important to notice the effect of inapplicability as the model detail (granulation) grows. The introduction of the term model lag in chapter 2 (See section 2.3.2) was introduced to counter the effect of the model not being applicable to all processes in reality. Actions that are deemed inapplicable to a certain Requirements Engineering process may be said to be Satisfied-Explained, thus enabling usage of the model in spite of this (See section 2.4). The risk is that as the model grows the Actions become more detailed (low-level), and this in turn can lead to an increased gap between the REPM model and most projects Requirements Engineering processes, i.e. greater model-lag.

The use of diagrams to graphically present the result of a project evaluation was considered before it was suggested by Mr. Cronberg. However his recommendation further anchored our intentions of using diagrams. On the other hand it is important to realize that the use of diagrams can be counterproductive. It may seem obvious that diagrams are to be used only when warranted, but in some cases diagrams are used as a rule. We have chosen to use several simple diagrams in a uniform way throughout the paper (See section 2.3.3).

Looking at the interview results under the section above one could get the impression that problems and inadequacies dominate the validation and the REPM model as a whole. This is not the case. The interview was made to validate the model, i.e. bring possible improvements to the surface. Due to this we have focused on the potential problems rather than the benefits. Most of the model went through the validation without any comments. This does not mean that these un-commented parts are perfect – rather good enough to suit their purpose in this stage of the model’s development. In addition the modification to the REPM model made as a result of the validation has significantly improved the quality and usability of the model. With this we feel confident that we can take the next step and test the model in a live setting, by using it to evaluate a number of real projects.
Chapter Four – Project Evaluation (REPM Model Validation Part II)

4 Chapter Four – Project Evaluation (REPM Model Validation Part II)

The purpose of this chapter is to present the second round of validation performed to validate the REPM model and the evaluation method presented in chapter two (See section 2.3). This was done through evaluating four different projects conducted by four different companies. In addition a pre-evaluation was done (before the project evaluation) to give us early feedback on the method used for the project evaluation.

This chapter is divided into three main parts. First, a description of how the project evaluation was prepared, structured, pre-evaluated and conducted. Second, the results of the project evaluations are presented with our analysis. Last we have a summary and conclusions.

4.1 Project Evaluation Method

In order to enable evaluation of four projects in industry the REPM evaluation checklist was constructed (See Appendix III). The goal of the project evaluation is to use the model in industry to validate the model as far as contents and structure are concerned (much like in the first validation (See chapter 3), but also to evaluate how well the checklist approach holds up, i.e. our way of putting the REPM model into action. No direct questions about the REPM model were posed to the members of industry, like in the first validation, but rather information about the model was gathered through the project evaluations. The REPM evaluation checklist was evaluated in the same way, i.e. by using it. In addition to conducting the project evaluations a general discussion about the REPM model and the use of it was encouraged and carried out after the project evaluations.

4.1.1 Method

The REPM evaluation checklist was used as a questionnaire of sorts – the result was structured interviews [22] as far as the project evaluations were concerned. In addition to this a discussion around the REPM model and the questionnaire gave us more in-depth data about what the interview subjects’ opinions pertaining to the model and how it was used for project evaluation.

A copy of the REPM Model (Version 0.95 – can be obtained from the authors) was sent to the interview subjects in advance. This to give all people involved a chance to study the model and its constituents in order to prepare them for the work during the project evaluations and to note any questions they wished to address.

In addition to this the participating companies were asked to choose a project to be evaluated and instructed that the project should be of a customer-developer type, i.e. not an internal development project but featuring an external customer. This due to the fact that the REPM model at this stage of its development is adapted primarily for these types
of projects. In addition we asked the person(s) mainly responsible for the Requirements Engineering process for the projects selected to be present as the interview subjects at the project evaluation session.

At the initial stage of the interview/evaluation the basic layout of the evaluation was clarified. Information about how the Project Evaluation Checklist version 1.4 was structured, and how it was connected to the REPM model was provided. In addition we asked the people evaluating the project to “think-aloud” when answering the questions. An important thing we clarified was the concept of Satisfied-Explained Actions, i.e. that the model may not suit all projects and that this concept was introduced to compensate for that fact.

All interviews were made on-site and in person. Two companies were situated in Sweden and two in Ireland. In person interviews are beneficial in several ways. It is often possible to extract more information as well as avoiding misunderstandings. But there can also be negative aspects like the interviewer influencing the interviewees [22] [25].

All interviews were documented using a digital camcorder. This to record every aspect of the interviews, mainly the discussions during and after the project evaluation, making in-depth analysis possible after the fact. Not because the questions themselves needed explanation but the dialogue around the checklist was important to remember. The interviews in Ireland were preceded with personal contact and emails to prepare for the evaluation (this correspondence can be obtained from the authors).

After the interviews were concluded and the material summarized, the interview subjects were offered to view the result before the material was published.

**4.1.2 Selecting the Companies and Interview Subjects**

For time reasons the selection was done using convenience sampling, i.e. we talked to companies that we had a previous relationship too. A random selection from all of the companies that fit the profile described above would unfortunately take too much time and demand resources that we did not have. The negative aspect of convenience sampling is that the selection method is not statistically satisfactory [25] [26]. However we did have a plan when selecting the companies. The idea was to have two large companies, i.e. more than 150 employees, and two small ones (>150 employees). This to ascertain that the model was tested on larger and smaller companies. The only criterion demanded from the companies selected was that they had projects featuring a customer-developer relationship.

The goal was to test projects with different levels of Requirements Engineering process maturity (REPM). We hoped that this would be possible but due to our sampling technique this was not certain beforehand. An additional risk is that one of the companies (and interview subjects) selected was present during the first validation (See chapter 3). We are aware of the risk that the subject in question could be influenced by this fact and that the results from the project evaluation conducted at this company could be influenced as well. However due to the subject’s professionalism the risk is very low.
Chapter Four – Project Evaluation (REPM Model Validation Part II)

The subjects being interviewed were in senior positions pertaining to the projects at hand and had forehand knowledge about Requirements Engineering, and more importantly extensive knowledge about the projects selected for evaluation. The project’s main responsible for Requirements Engineering was designated “project responsible” for each project evaluation session, but we did not put any limitations on the number of people that were present. We feel that the positive effects of having a discussion with more than one person in a senior position outweigh the risks.

Each of the projects being evaluated was selected by the interview subjects. This made it possible for the companies to avoid questions dealing with a project very sensitive to being exposed to outside parties and also made it possible for them to choose a project that was concluded, of the right sort (customer-developer) and of interest to get evaluated. However this way of choosing projects is prone to risk due to several reasons. The sampling is very much tainted by the person choosing the project and it may not be representative for the company at hand - it is a case of convenience sampling. In addition the choosing party can be biased, i.e. trying to portray the company in a positive way. However for our primary purposes it does not matter if the project in question was representative. We did not set out to evaluate companies, but rather the REPM model itself.

The reason for interviewing companies both in Sweden and Ireland was to get an impression if the model was suitable for use not only nationally but internationally, and to get an idea of the differences between companies in Sweden and Ireland pertaining to Requirements Engineering.

4.2 Evaluation of REPM Checklist

In order to validate the Project Evaluation Checklist – checking its content, structure and formulation we chose to invite an uninitiated person, a student on the Master Program for Software Engineering, to go through the checklist and evaluate a project she had been in charge of. While going through the checklist we asked her to “think-aloud” and comment on her reasoning and how she understood the questions. We logged the process noticing what questions needed clarification. We gave her a copy of the checklist and the REPM model, and explained basic things, i.e. the UID column (See Appendix III) and that she was free to use the model if she felt the need for clarification on any question.

4.2.1 Initial Test result

The project evaluation with the student took about two hours and afterwards we discussed the model and the checklist. She understood approximately 30% of the questions directly, another 40% as she consulted the REPM model while reading the question, and about 30% of the questions were difficult to understand, i.e. she misunderstood the questions and/or did not understand them at all. Ill formulated questions and too brief questions were the main problems here. Another problem was that the student was not well versed in the vocabulary of Requirements Engineering (she was
not a requirements engineer). For this reason we opted not to change all the things she did not understand. All other problems were dealt with.

4.3 Project Evaluation

For reasons of anonymity the names of the projects below are changed, i.e. not revealing what company corresponds to which project.

Below we present diagrams for each of the MPAs (Main Process Area) together with a short description. In addition the Actions categorized to be Satisfied-Explained by the project responsible (the person doing the evaluation) are presented and the reason for putting them into this category is given. Under every graph (denoting each MPA) we have three subsections, “General”, “Improvements” and “Improvement consequences”. General denotes facts extracted from the diagram. Under Improvements we have suggestions about what parts of the Requirements Engineering process could be improved, i.e. what Actions could be completed additionally. Improvement consequences are the effects to the project/company that the suggested improvements could have.

A project specific summary/discussion of the results is presented last under each section. In combination with every instance of Satisfied-Explained a marker is present in the form of a comment id, e.g. *COMMENT ID: 1*. This to cross-reference our analysis to the relevant Action when necessary.

During the evaluation we refer to the names of the REPM levels as follows:
REPM level 1 = Wood
REPM level 2 = Bronze
REPM level 3 = Silver
REPM level 4 = Gold
REPM level 5 = Platinum

The reason for presenting all of the projects in detail below is that they are real examples of how an evaluation with the model is to be performed and how the result of such an evaluation can be presented. The details also show examples of how to interpret an evaluation of a project. The presentations of the four projects cover most examples of results, improvements, consequences and comments.

Evaluating projects from two different countries (Ireland and Sweden) was done to show the model’s applicability to projects internationally. Geographical location should not be a limitation when it comes to using the REPM model for evaluation of a project’s Requirements Engineering process.
4.3.1 Project Evaluation One – Project Alpha

The result of the REPM project evaluation is presented below and referred to as “Project Alpha”. Project Scope was 11000 man-hours and entailed the delivery of a partially finished and standardized system solution aimed towards logistics and warehouse management. The company (and all of its customers) is highly specialized in the domain at hand. In addition the customer in Project Alpha has fairly extensive system knowledge, i.e. know what they need and want due to extensive domain knowledge and the use of similar systems before.

4.3.1.1 MPA of Requirements Elicitation

Total number of Elicitation Actions on each REPM level compared with the amount of Elicitation Actions completed and Satisfied-Explained in Project Alpha.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 4.1  Project Alpha - chart 1.
Chapter Four – Project Evaluation (REPM Model Validation Part II)

**General**
From this chart we can ascertain that Project Alpha resides on REPM level 0 when it comes to the MPA of *Elicitation*. In the MPA of *Elicitation* you can note that only two Actions need to be completed in order for the project to reside on *Silver*. Furthermore it is interesting to notice that the *model lag* is fairly low within this MPA with only three Actions in the category Satisfied-Explained. The Actions, and the project’s motivation for excluding them, are:

**E.1.a2 Research Stakeholders**
The customer is given the task of identifying the potential stakeholders and doing the relevant research. They are much more aware of who the stakeholders are and what the relevant groups are. *COMMENT ID: 1*

**E.2.a2 General Stakeholders**
To consult all of general stakeholders takes too much time. However it is not only the executive stakeholders that are consulted but also people at middle management level that have often worked in the company a long time and are familiar with the domain and current systems. *COMMENT ID: 2*

**E.3.a1 Human Domain Consideration**
Usually the people ordering the system are the ones pushing the possible political and organizational factors that might influence the rest of the workers. Therefore the information from lower levels is not always considered. It is not people on the lower levels that make the decisions and it would not make for a good business solution to involve them. *COMMENT ID: 3*

**Improvements**
In order to advance to *Silver* the Actions *E.4.a3 In-house Scenario Creation* and *E.2.a3 In-house Stakeholders* have to be completed.

In-house stakeholders are not consulted during the Requirements Engineering process, nor are there any in-house scenarios created. To complete the two Actions the in-house stakeholders, i.e. the developers, should be involved in the Requirements Engineering process.

**Improvement consequences**
Not involving in-house stakeholders in the Requirement Engineering process can in some cases be understandable and one has to take company culture into consideration, i.e. the reason may have its roots in political and organizational factors. However utilizing an already present resource (like developers) can be beneficial. The developers often possess knowledge that can be used to further the Requirements Engineering process and clarify the requirements - an additional viewpoint is introduced. Furthermore creating scenarios may help the project, i.e. better solutions and clearer requirements, but may also help the developers. If the in-house stakeholders are a part of the process they can better understand what a certain requirement is really about, i.e. what the underlying reason for it is, and thus be more effective in implementing it later. Reaching *Silver* will improve the quality and refinement of the requirements by using already present resources.
4.3.1.2 MPA of Requirements Analysis and Negotiation

Total number of Analysis and Negotiation Actions on each REPM level compared with the amount of Analysis and Negotiation Actions completed and Satisfied-Explained in Project Alpha.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that Project Alpha resides on REPM level 0 but still several Actions are completed (all on REPM level 2). Only one Action needs to be added in order to reach Bronze and an additional two to reach Silver.

Further it is of interest to notice that the model lag is nonexistent under this MPA.
Improvements
In order to reach Silver three additional Actions have to be completed - A.a1 Analysis Through Checklists on REPM level 1, A.a3 Interaction Matrices and A.3.a3 Risk Assessment – selected on REPM level 3.

There was no official use of checklists during this process but there was an unofficial use of checklists up to the individual developer. By formalizing and standardizing the use of checklists one could take a non-mandatory practice and make it standardized and official. Interaction matrices and risk assessment are areas that are not done either. Risk assessment should be carried out on volatile requirements, i.e. the ones prone to change, this is not done at all today but according to the project responsible it should be. The use of interaction matrices maps the dependencies between requirements.

Improvement consequences
Some improvements can easily be made by just documenting certain tasks that are present unofficially, e.g. in the analysis process official checklist could be used. This would enable a standardized way of analyzing the requirements throughout the organization and the project in question. This may also alleviate the risk that some parts of an analysis of a certain requirement are overlooked.

Using interaction matrices dependencies, both negative and positive, could be mapped and taken into consideration. This could prove beneficial in many ways, e.g. information about problems (one requirement affecting/contradicting another), and positive effects (dependencies making for one solution for two requirements etc.).

In the case of risk assessment of selected (volatile) requirements the benefits are that one can take future potential problems into consideration, and plan for them. It is not easy to evaluate risks, but the benefit of doing a simple try can often make a lot of difference, i.e. things you did not think of come up and you specify the potential “problem requirements” and why you consider them to be volatile. This act can by itself make you aware of, and sensitive to, potential problems.

Reaching Silver will help take issues like requirements dependencies and risks into consideration, as well as standardizing and articulating the analysis process.
4.3.1.3 MPA of Requirements Management

Total number of Management Actions on each REPM level compared with the amount of Management Actions completed and Satisfied-Explained in Project Alpha.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that Project Alpha resides on REPM level 1 when it comes to the MPA of Management. The jump to REPM level 2 is fairly large. Furthermore it is interesting to notice that the model lag is fairly low within this MPA with only one Action in the category Satisfied-Explained The Action, and the project’s motivation for excluding it, is:

**Figure 4.3** Project Alpha - chart 3.

**General**

From this chart we can ascertain that Project Alpha resides on REPM level 1 when it comes to the MPA of Management. The jump to REPM level 2 is fairly large. Furthermore it is interesting to notice that the model lag is fairly low within this MPA with only one Action in the category Satisfied-Explained The Action, and the project’s motivation for excluding it, is:
Chapter Four – Project Evaluation (REPM Model Validation Part II)

**M.1.3.a1 Prototyping**
This is not done due to that prototypes are not considered to reflect the adaptation of the system but rather individual things. Prototypes also pose a threat for the developers since the customer might mistake prototypes for a finished system component and not mock-ups. *COMMENT ID: 4*

**Improvements**
In order to reach Silver five additional Actions have to be completed:
- **M.1.2.a2 Quantitative Requirement Description** (REPM level 2)
- **M.1.4.a4 User Manual Draft** (REPM level 2)
- **M.3.a4 Backward-to traceability** (REPM level 2)
- **M.1.4.a2 Requirements Review** (REPM level 3)
- **M.4.a1 Volatile Requirements Identification** (REPM level 3)

Drafting user manuals at the requirements stage and official requirements reviews can be done to validate the requirements. Identifying volatile requirements can be used to recognize requirements prone to change at an early stage and incorporate this information in the overall Requirements Engineering process and/or the general project planning process. To be able to trace implemented components, e.g. classes and procedures, to the requirements they stem from can further understanding of why a certain thing is implemented. Quantification of non-functional requirements furthers the level of requirements specification (more information about a requirement) and improves testability. Not quantifying requirements can have serious repercussions. If a client assumes that the uptime will be e.g. 99.995 and the best you can achieve is 98.5 there can be a problem. Specifying 98.5 and showing this to the client catches this problem early on.

**Improvement consequences**
Validation of requirements through formal peer-reviews can give new views to problem areas, ensure that standards and rules are followed and help further refine requirements. The positive effect of having external (peers) reviewers is that they can help to find problems the internal developers overlooked. Formal reviews are often contractually bound and this fact can instill confidence in the project from the customers point of view. Reviews can be used as an official measurement of quality.

Drafting user manuals that stem from the requirements can also be a tool for validation. User manuals tend to see things from the end-users perspective and is often based in scenarios and/or action-reaction events. Many unforeseen things may become evident when going through this process, e.g. if a user is to save something and “saving” is not a requirement, then you know that there may be a problem. The identification of volatile requirements at an early stage may help alleviate the risk of surprise if a requirement suddenly changes. A requirement prone to change may be the cause of many potential problems, e.g. other requirements are dependent on the volatile one. If you are aware of the potential problems you can plan for them and avoid damaging consequences.

Reaching Silver will help in ensuring that requirements are refined and validated.
4.3.1.4 Action Summary

Total number of Actions on each REPM level compared with the amount of Actions completed and Satisfied-Explained in Project Alpha.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

General

From this chart we can ascertain that Project Alpha resides on REPM level 0, i.e. all of the Actions on REPM level 1 are not satisfied. Project Alpha follow the general curve and the model lag is moderate to fairly small, i.e. four Actions in total.

4.3.1.5 Summary/Discussion

Customer relations in this project are based on a customer-developer paradigm but with some modification. In the case of Project Alpha the customer is very well versed in what they needed and how they operate, i.e. they have been established for a fairly long time and already have systems (computerized and otherwise) supporting their processes. Project Alpha is a contract to deliver a replacement solution. Furthermore the solution
being delivered is based on a general system suit that is sold to several customers. The
development (which is based on the Requirements Engineering being evaluated here) is
of a customer adaptation sort, i.e. certain things are adapted to suite the individual
customer. The core of the system is however the same for all customers.

The fact that the developers on Project Alpha are unusually knowledgeable when it
comes to the domain makes it easier for the customer and the developers to interact and
speak the same language. The elicitation of requirements can be easier and the risk for
misunderstandings can be less. The knowledge that the ones with the pertinent data are
situated in middle management can be a time and effort saver, i.e. knowing who to talk to
and get the correct answers (*COMMENT ID: 2*). On the other hand there is a risk that
some things are taken for granted and thereby missed due to lack of scrutiny. This can
also be true for not researching who the stakeholders are, leaving this up to the customer
(*COMMENT ID: 1*). However, this kind of research can be costly and the risk may be
acceptable. The same may be said for the trust placed in the customer when it comes to
taking human domain considerations into account. Political factors are often present and
can be a source of misinformation, e.g. the official “story” comes out, but not the truth.
This is however alleviated in the case of Project Alpha due to the fact that they only
consult middle management and these representatives are, according to the project
responsible, seldom influenced by upper management in the same way as a lower level
general stakeholder would be (*COMMENT ID: 3*).

Prototyping can be a hazardous undertaking and may involve risks like the customer
taking prototypes for finished system components; this is taken into consideration in the
REPM model as well. This does not mean that the construction of prototypes is a bad
idea in general. They can be used effectively to validate and/or clarify requirements and
are easy to understand by customers. The fact that a customer takes the prototype for a
finished system component is a question of maturity and communication. In the case of
Project Alpha the customer’s maturity and domain knowledge is high, therefore the real
meaning of prototypes should be possible to convey through adequate communication
(*COMMENT ID: 4*).

The REPM model is designed to make evaluation of the Requirements Engineering
process possible. Through this evaluation some ideas for improvement may become
apparent. Studying the graphs and data for Project Alpha, it seems as if the maturity of
the project’s Requirements Engineering process is such that it would be the most
beneficial improvement step to fulfil the Silver REPM level. This would ensure
improvements in many areas and also represent a consistent, coherent and well thought-
through Requirements Engineering process. Traceability in the project pertaining to
requirements, validation of requirements, templates for ensuring standardized ways of
doing analysis, and taking advantage of in-house competence is some of the areas that
will be added if Silver is achieved. Therefore, our improvement suggestions above are
focused on achieving this level of maturity. What parts should and should not be
modified/implemented is up to the organization at hand due to the fact that they are
intimately familiar with their own situation.
4.3.2 Project Evaluation Two – Project Beta

The result of the REPM project evaluation is presented below and referred to as the “Project Beta”. Project Scope was approximately 18000 man-hours and entailed developing an extensive application support system for a government agency. The system was based on a previous mainframe-style system but was downsized, modernized (redone and object orientated) and a GUI was added.

4.3.2.1 MPA of Requirements Elicitation

Total number of *Elicitation* Actions on each REPM level compared with the amount of *Elicitation* Actions completed and Satisfied-Explained in Project Beta.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

*Figure 4.5*  Project Beta - chart 1.
Chapter Four – Project Evaluation (REPM Model Validation Part II)

**General**
From this chart we can ascertain that Project Beta resides on REPM level 5, i.e. *Platinum* when it comes to the MPA of *Elicitation*. Furthermore it is interesting to notice that the *model lag* is fairly low under this MPA with only two Actions in the category Satisfied-Explained. The Actions, and the project’s motivation for excluding them, are:

**E.4.a2 Scenario Elicitation - General Stakeholders**
The General Stakeholders do not know enough about the system/domain to answer the relevant questions for scenario creation. In addition the information needed from these stakeholders comes through informal channels. *COMMENT ID: 5*

**E.1.a2 Research Stakeholders**
For political reasons no stakeholder research is done officially. Unofficially some mapping of the groups is conducted. *COMMENT ID: 6*

**Improvements**
*Platinum* is achieved.

**Improvement consequences**
*Platinum* is achieved.
4.3.2.2 **MPA of Requirements Analysis and Negotiation**

Total number of *Analysis and Negotiation* Actions on each REPM level compared with the amount of *Analysis and Negotiation* Actions completed and Satisfied-Explained in Project Beta.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**General**

From this chart we can ascertain that Project Beta resides on REPM level 3, i.e. *Silver* when it comes to the MPA of *Analysis and Negotiation*. According to the chart (Figure 4.6) only one Action has to be completed to advance to *Platinum*. This is not the whole truth in this case however, due to the fact that the Action not completed belongs to an
optional group (OG1). Due to that other Actions in the optional group are completed formally the one not completed here, i.e. the one missing in the diagram (A.3.a1 Risk Assessment – individual) does not hinder the project to be present on REPM level 5 (Platinum). At present the presentation (diagrams) do not depict optional Actions and/or optional groups of Actions. Consideration to this is paid in the text (like above).

Furthermore it is interesting to notice that the model lag is fairly low under this MPA with only one Action in the category Satisfied-Explained The Action, and the project’s motivation for excluding it, is:

**A.a3 Interaction Matrices**
Not done, the dependencies plotted here are often incorrect. *COMMENT ID: 7*

**Improvements**
*Platinum* is achieved.

**Improvement consequences**
*Platinum* is achieved.
4.3.2.3 MPA of Requirements Management

Total number of Management Actions on each REPM level compared with the amount of Management Actions completed and Satisfied-Explained in Project Beta.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 4.7** Project Beta - chart 3.

**General**

From this chart we can ascertain that Project Beta resides on REPM level 0, i.e. not all of the Actions at REPM level 1 are completed pertaining to the MPA of Management.

Furthermore it is interesting to notice that the model lag is fairly low under this MPA with only two Actions in the category Satisfied-Explained The Actions, and the project’s motivation for excluding them, are:

**M.1.1.a1 Document Summary**

The requirements are written in summaries as is, there is no need to summarize them - rather clarify and expand them. *COMMENT ID: 8*
Chapter Four – Project Evaluation (REPM Model Validation Part II)

M.1.2.a1 Requirements Description Template
The customer has many different requirements stemming from different areas. Therefore it is not beneficial to use the same template for all of the requirements due to the fact that they are different, stemming from different areas and ought to be described differently.

*COMMENT ID: 9*

Improvements
In order to reach Platinum eight additional Actions have to be completed;

M.1.1.a4 Term Definition (REPM level 1)
M.1.1.a2 Document Usage Description (REPM level 2)
M.3.a4 Backward-to traceability (REPM level 2)
M.1.a1 Record Requirements Rationale (REPM level 3)
M.1.3.a2 System Models (REPM level 3)
M.3.a3 Forward-from traceability (REPM level 3)
M.1.4.a3 System Model Paraphrasing (REPM level 5)
M.3.a6 Version traceability (REPM level 5)

Improvement consequences
The requirement document is important, all results from the Requirements Engineering process are presented here and the document is read by many different groups, e.g. developers, managers and sometimes customers. In addition the people belonging to a certain group often have a unique background. For the document to be as readable (usable) as possible two things can be included, a term definition and a usage description. The term definition helps different users understand specific terms, everything from domain specific expressions to technical terms. While the usage description gives an explanation of how the requirements document should be used/read by different user groups in order to offer optimal benefit. Together the usability of the requirements document is enhanced and time is saved, i.e. there will be fewer questions about what a certain term means, and you only read what you need.

Requirements rationale anchors a certain requirement to reality, i.e. contains information about why the requirement was specified in the first place and what function it has. This gives the readers/users of the requirement document vital background information about a requirement and prevents the initial meaning of the requirement to be lost during the Requirement Engineering process.

Traceability is important and offers many benefits. Linking the requirements to the design (and indirectly to the implemented components) makes it possible to validate that a certain requirement has in fact been implemented. The other way around, i.e. linking the design and implementation components to the requirements enables developers to find out the reason for why a certain item is present. This can also be used as a form of validation, i.e. a developer can go back and really make sure that all aspects of a requirement are taken into consideration as he/she implements an item (is every necessary part implemented, are there dependencies between implemented components that have been missed during Requirements Engineering etc). Traceability also includes version handling when it comes to individual requirements. Many have versions of the
whole of the requirements document and/or versions of specific critical requirements, but
it can also be beneficial to handle different versions of all of the specific requirements.
The ability to virtually go back and see the progress of a requirement during its
development can be used for presentation purposes, i.e. to show stakeholders the
evolution of a requirement. It can also be used by developers to gain understanding of
why a certain thing has changed. In addition requirements that are to be reused can be
captured at different stages of evolution.

There are many ways of validating requirements. One that can be fairly effective (but
also time consuming) is the act of paraphrasing models created for a requirement.
Starting from the models and trying to write down the requirement in text is a sort of
reverse engineering. When done the text created can be compared with the one acting as
the base for the model in the first place. This enables a rather effective validation to take
place and the requirement can be clarified and checked for lapses, i.e. the model(s) are
checked and thereby the requirement.

Domain knowledge is crucial to a successful Requirements Engineering process. The use
of system models to learn more about (take consideration to) the environment can be a
powerful tool in learning more about the domain. An example of a system model can be a
stimulus-response model created to simulate an environment. Many system models are
not directly associated with a certain requirement, rather indirectly. The ability to ask the
right questions correlates to how effectively you can take the environment into
consideration, i.e. the environment in which the system being created is to exist. When
your understanding of a domain grows your ability to create requirements that interact
with it in a positive manner also grows.
4.3.2.4 Action Summary
Total number of Actions on each REPM level compared with the amount of Actions completed and Satisfied-Explained in Project Beta.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that Project Beta resides on REPM level 0, i.e. all of the Actions on REPM level 1 are not satisfied. Project Beta follows the general curve and the model lag is moderate to fairly small, i.e. five Actions in total. Nine additional Actions have to be completed in order to reach Platinum.

4.3.2.5 Summary/Discussion
Project Beta is conducted with a government agency as the customer. This can be a special case in reference to the customer-developer relationship. Influencing factors like regulations and practices as well as human domain factors, i.e. political factors, can...
influence the Requirement Engineering process. In the case of Project Beta this is clearly illustrated through the inability to do an official examination of who all of the stakeholders are (*COMMENT ID: 6*). Certain organizations are rigid in their acceptance of arguments for doing a certain Action and many things have to be done through unofficial channels (*COMMENT ID: 5*). Basically it is up to the requirements engineer to recognize the problem and take appropriate action. The problem in the case of Project Beta is that some stakeholders may have been overlooked due to the fact that some of the information gathering was done through back channels. On the other hand it is commendable that the Action in question (E.1.a2 Research Stakeholders) was not ignored altogether.

Replicability, i.e. to be able to repeat a process in the same way, is crucial in an engineering environment. The use of templates for different purposes helps ensure this. In the case of Project Beta no template was used for describing the requirements. The necessity to specify different requirements in different ways is understandable, but most requirements have some basic things in common. This can be such simple things as unique identifiers, naming standards for every requirement and general plain text descriptions. These things all requirements have in common can constitute the base of a general requirements description template (*COMMENT ID: 9*). This general template can be further specialized into several sub-templates, each used for specifying requirements of a certain type.

Using interaction matrices dependencies, both negative and positive, could be mapped and taken into consideration. This could prove beneficial in many ways, e.g. information about problems (one requirement affecting/contradicting another), and positive effects (dependencies making for one solution for two requirements etc.). To create an accurate and adequate interaction matrix can be a large task, and in many cases the dependencies showed are faulty. However the act of simply creating a matrix and trying to structure dependencies may be enough to notice problems/benefits not recognized earlier (*COMMENT ID: 7*).

In general the MPA of Requirements Management is the weak link in Project Beta’s Requirements Engineering process. Through this evaluation some ideas for improvement may become apparent. Studying the graphs and data for Project Beta, it seems as if the maturity of the project’s Requirements Engineering process is such that it would be the most beneficial improvement step to fulfil the Platinum REPM level. This would ensure improvements in many areas and also represent a consistent, coherent and well thought-through Requirements Engineering process. Readability (and usability) of the requirements document, advanced traceability in the project pertaining to requirements, and formal and extensive validation of requirements are some of the areas that will be added if Platinum is achieved. Therefore, our improvement suggestions above are focused on achieving this level of maturity. What parts should and should not be modified/implemented is up to the organization at hand due to the fact that they are intimately familiar with their own situation.
4.3.3 Project Evaluation Three – Project Gamma

The result of the REPM project evaluation is presented below and referred to as “Project Gamma”. Project Gamma is a web based system designed for a government agency. The project is the development of a product that assists companies that are FDA regulated to capture data that will aid in the continued upkeep of their FDA status. The man-hours for the Requirements Engineering process was 350.

4.3.3.1 MPA of Requirements Elicitation

Total number of Elicitation Actions on each REPM level compared with the amount of Elicitation Actions completed and Satisfied-Explained in Project Gamma.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that Project Gamma resides on REPM level 0 when it comes to the MPA of Elicitation. Furthermore it is interesting to notice that the model lag...
is fairly low under this MPA with only two Actions in the category Satisfied-Explained. The Actions, and the project’s motivation for excluding them, are:

**E.3.a4 Business Domain Consideration**
The System being developed is to be sold to a third party by our customer.  
*COMMENT ID: 10*

**E.a1 Requirements Reuse**
The system being developed is unique for the organization at hand, i.e. no other similar systems have been created in the same application domain and thus there is nothing to reuse.  
*COMMENT ID: 11*

**Improvements**
In order to reach **Silver** two additional Actions have to be completed;

*E.4.a3 In-house Scenario Creation (REPM level 1)*
*E.4.a1 Scenario Elicitation – Executive Stakeholders (REPM level 2)*

**Improvement consequences**
Not involving in-house stakeholders, e.g. developers, designers and managers, in the Requirement Engineering process can in some cases be understandable and one has to take company culture into consideration, i.e. the reason may have its roots in political and organizational factors. Utilizing an already present resource however (like developers) can be beneficial. The developers often possess knowledge that can be used to further the Requirements Engineering process and clarify the requirements, an additional viewpoint is introduced. Furthermore creating scenarios may help the project, i.e. better solutions and clearer requirements, but may also help the developers. If the in-house stakeholders are a part of the process they can better understand what a certain requirement is all about really, i.e. what the underlying reason for it is, and thus be more effective in implementing it later. The same goes for eliciting scenarios from the executive stakeholders, i.e. the ones ordering the system. Getting their view/feedback can be beneficial at the early stage of requirement elicitation.

Reaching **Silver** will improve the quality and refinement of the requirements by using already present resources inside the company, as well as resources outside committed to the project.
4.3.3.2 MPA of Requirements Analysis and Negotiation

Total number of Analysis and Negotiation Actions on each REPM level compared with the amount of Analysis and Negotiation Actions completed and Satisfied-Explained in Project Gamma.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that Project Gamma resides on REPM level 0 when it comes to the MPA of Analysis and Negotiation. According to the chart (Figure 4.10) only four Actions have to be completed in order for the project to advance to Silver. Furthermore it is interesting to notice that the model lag is non-existent under this MPA.
Chapter Four – Project Evaluation (REPM Model Validation Part II)

**Improvements**

In order to reach Silver four additional Actions have to be completed;

*A.a1 Analysis Through Checklists (REPM level 1)*
*A.a3 Interaction Matrices (REPM level 3)*
*A.1.a1 Boundary definition through categorization (REPM level 3)*
*A.3.a3 Risk Assessment – selected OG1.03 (REPM level 3)*

**Improvement consequences**

Some improvements can easily be made by documenting certain tasks that are present unofficially, e.g. in the analysis process official checklists could be used. This would enable a standardized way of analyzing the requirements throughout the organization and the project in question. This may alleviate the risk that some parts of an analysis of a certain requirement are overlooked, i.e. if the list is followed the contents of the list (template) dictates what is to be done.

Using interaction matrices dependencies, both negative and positive, could be mapped and taken into consideration. This could prove beneficial in many ways, e.g. information about problems (one requirement affecting/contradicting another), and positive effects (dependencies making for one solution for two requirements etc.).

To evaluate if a requirement is within the boundary of the system it can be beneficial to categorize the requirements into different groups, for example the following groups; System Requirements (inside the system scope), Requirements for the operational processes associated with the system and Requirements clearly outside the scope. Operational processes requirements are defined by the fact that they are in need of information that lies outside the scope of the designed database of the system and subsequently in need of human decision making.

In the case of risk assessment of selected (volatile) requirements the benefits are that one can take future potential problems into consideration, and plan for them. It is not easy to evaluate risks, but the benefit of doing a simple try can often make a lot of difference, i.e. things you did not think of come up and you specify the potential “problem requirements” and why you consider them to be volatile. This act can by itself make you aware of, and sensitive to, potential problems.

Reaching Silver will help take issues like requirements dependencies and risks into consideration, as well as standardizing and articulating the analysis process.
4.3.3.3 MPA of Requirements Management

Total number of Management Actions on each REPM level compared with the amount of Management Actions completed and Satisfied-Explained in Project Gamma.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that Project Gamma resides on REPM level 1, i.e. Wood, in the MPA of Management.

Furthermore it is interesting to notice that the model lag is fairly low under this MPA with only two Actions in the category Satisfied-Explained. The Actions, and the project’s motivation for excluding them, are:

**Figure 4.11** Project Gamma - chart 3.

**General**

From this chart we can ascertain that Project Gamma resides on REPM level 1, i.e. Wood, in the MPA of Management.

Furthermore it is interesting to notice that the model lag is fairly low under this MPA with only two Actions in the category Satisfied-Explained. The Actions, and the project’s motivation for excluding them, are:
M.4.a1 Volatile Requirements Identification
No attention is paid to identifying volatile requirements because the customers can change the requirements as many times they want, it does not matter. For every change the contract (how much time and money it takes to complete the project) is renegotiated. *COMMENT ID: 12*

M.1.3.a3 Environmental Models
The system being developed is a “stand-alone” system, i.e. the environment does not have any influence on it. *COMMENT ID: 13*

Improvements
In order to reach Silver eight additional Actions have to be completed:

M.1.1.a2 Document Usage Description (REPM level 2)
M.1.2.a1 Requirements Description Template (REPM level 2)
M.1.2.a2 Quantitative Requirements Description (REPM level 2)
M.1.3.a1 Prototyping (REPM level 2)
M.1.4.a4 User Manual Draft (REPM level 2)
M.3.a2 Backward-from traceability (REPM level 2)
M.1.1.a3 Business Case (REPM level 3)
M.1.4.a2 Requirements Review (REPM level 3)

Improvement consequences
The requirement document is important, all results from the Requirements Engineering process are presented here and the document is read by many different groups, e.g. developers, managers and sometimes customers. In addition the people belonging to a certain group often has a unique background. For the document to be as readable (usable) as possible a usage description can be added. The usage description gives an explanation of how the requirements document should be used/read by different user groups in order to offer optimal benefit and the different user groups only read what they need.

Replicability, i.e. to be able to repeat a process in the same way any number of times, is crucial in an engineering environment. The use of templates for different purposes helps ensure this. The necessity to specify different requirements in different ways is understandable, but most requirements have some basic things in common. This can be such simple things as unique identifiers, naming standards for every requirement and general plain text descriptions, e.g. how to write the description and what should be included. The things all requirements have in common can constitute the base of a general requirements description template. This general template can be further specialized into several sub-templates, each used for specifying requirements of a certain type.

The act of quantifying requirements is a form of refinement, i.e. it gives more information about what is needed and should be done. Not quantifying requirements can have serious repercussions - if a client assumes that the uptime will be e.g. 99.995 and the best you can achieve is 98.5 there can be a problem. Specifying 98.5 and showing this to
the client catches this problem early on. In addition the quantification of requirements enhances testability of the requirements in question.

Prototypes can be used effectively to validate and/or clarify requirements and are easy to understand by customers. To make prototypes at an early stage may help catch problems and inaccuracies early – in contrast to finding out problems at a later stage when design is started. Prototypes can be very simple, a user interface, but can be a powerful tool in exchanging information. Drafting user manuals that stem from the requirements can also be a tool for validation. User manuals tend to see things from the end-users perspective and is often based in scenarios and/or action-reaction events. Going through this process many unforeseen things may become evident, e.g. if a user is to save something and “saving” is not a requirement, then you know that there may be a problem.

Validation of requirements through formal peer-reviews can give new views to problem areas, ensure that standards and rules are followed and help further refine requirements. The positive effect of having external (peers) reviewers is that they can help to find problems the internal developers overlooked. Formal reviews are often contractually bound and this fact can instill confidence in the project from the customers point of view. Reviews can be used as an official measurement of quality.

Traceability is important and offers many benefits. To be able to trace where a requirement originates from enables developers to go directly to the source. This saves time, i.e. not having to track down the source of a particular requirement if there are questions.

Domain knowledge is crucial to a successful Requirements Engineering process. The use of system models to learn more about (take consideration to) the environment can be a powerful tool in learning more about the domain. An example of a system model can be a stimulus-response model created to simulate an environment. Many system models are not directly associated with a certain requirement, rather indirectly. The ability to ask the right questions correlates to how effectively you can take the environment into consideration, i.e. the environment in which the system being created is to exist. When your understanding of a domain grows your ability to create requirements that interact with it in a positive manner also grows.
Chapter Four – Project Evaluation (REPM Model Validation Part II)

4.3.3.4 Action Summary
Total number of Actions on each REPM level compared with the amount of Actions completed and Satisfied-Explained in Project Gamma.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**General**
From this chart we can ascertain that Project Gamma resides on REPM level 0, i.e. all of the Actions on REPM level 1 are not satisfied. Project Gamma follows the general curve and the model lag is moderate to fairly small, i.e. four Actions in total.

4.3.3.5 Summary/Discussion
The system being created through Project Gamma is situated in an environment in which it has very little interaction with other systems. A simple interface connecting it to one other part is the only exception. Creating environmental models may seem unnecessary from this standpoint. However the system still has an environment that may be beneficial...
to take into account. The environment covers not only other systems but business processes, i.e. all things happening outside of the system being created. Understanding the domain and the environment makes for better system production (*COMMENT ID: 13*). This is also true for understanding the business domain of the system. Understanding how the system being created will make a contribution to the organization for which it is being developed will help you to drive the elicitation process forward when having higher goals to strive for. System creation should be a cooperative process where value has a central role, i.e. making a system as good as possible (within the constraints posed), and this is up to the developing organization as much as it is up to the customer (*COMMENT ID: 10*).

Identifying volatile requirements can be very useful. Taking potential future problems into consideration and identifying where those problems could occur prepares you for action – rather than provoking a reaction. Requirements are often dependent on each other. This can take any number of forms, the extreme being total dependency. If you know what requirements may be volatile it is possible to foresee repercussions caused by change to one or several requirements (*COMMENT ID: 12*).

In general the MPA of *Requirements Management* is the weak link in Project Gammas Requirements Engineering process. Attention should also be paid to the MPA of *Analysis and Negotiation*.

Studying the graphs and data for the project, it seems as if the maturity of the project’s Requirements Engineering process is such that it would be the most beneficial improvement step to fulfil the *Silver REPM* level. This would ensure improvements in many areas and also represent a consistent, coherent and well thought-through Requirements Engineering process.

Traceability in the project pertaining to requirements, validation of requirements, templates for ensuring standardized ways of doing analysis and specifications, and taking advantage of in-house competence is some of the areas that will be added if *Silver* is achieved. In addition the introduction of risk management will prepare for possible problems. Therefore, our improvement suggestions above are focused on achieving this level of maturity. What parts should and should not be modified/implemented is up to the organization at hand due to the fact that they are intimately familiar with their own situation.
4.3.4 Project Evaluation Four – Project Delta

The result of the REPM project evaluation is presented below and referred to as “Project Delta”. Project Delta is a web based system designed to manage training for a specific Medical device company. The system took ten 1600 man-hours to develop.

4.3.4.1 MPA of Requirements Elicitation

Total number of Elicitation Actions on each REPM level compared with the amount of Elicitation Actions completed and Satisfied-Explained in Project Delta.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that Project Delta resides on REPM level 5, i.e. Platinum when it comes to the MPA of Elicitation. Furthermore it is interesting to notice that the model lag is fairly low under this MPA with only one Action in the category Satisfied-Explained. The Action, and the project’s motivation for excluding it, is:

**Figure 4.13** Project Delta - chart 1.
E.1.a2 Research Stakeholders
This system is a small system being tailor-made for a company. A specific case where the customers are close to the developers and the definition of who the stakeholders are is clear from the beginning. *COMMENT ID: 14*

**Improvements**
*Platinum* is achieved.

**Improvement consequences**
*Platinum* is achieved.
4.3.4.2 MPA of Requirements Analysis and Negotiation

Total number of Analysis and Negotiation Actions on each REPM level compared with the amount of Analysis and Negotiation Actions completed and Satisfied-Explained in Project Delta.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

General

From this chart we can ascertain that Project Delta resides on REPM level 0, i.e. not all of the Actions at REPM level 1 are completed pertaining to the MPA of Analysis and Negotiation.

Furthermore it is interesting to notice that the model lag is fairly large under this MPA with five Actions in the category Satisfied-Explained. They are:

Figure 4.14 Project Delta - chart 2.
Chapter Four – Project Evaluation (REPM Model Validation Part II)

A.2.a1 Prioritizing Requirements
The system is very small and there are few requirements, and all are critical. Since similar systems (known technology) have been developed before the developers know what to expect. *COMMENT ID: 15*

A.2.1.a1 Re-prioritization - New Requirements
See A.2.a1 Prioritizing Requirements *COMMENT ID: 16*

A.2.1.a2 Re-prioritization - New Releases
See A.2.a1 Prioritizing Requirements *COMMENT ID: 17*

A.2.1.a3 Re-prioritization with Regularity
See A.2.a1 Prioritizing Requirements *COMMENT ID: 18*

A.2.1.a4 Re-prioritization due to Change
See A.2.a1 Prioritizing Requirements *COMMENT ID: 19*

Improvements
In order to advance to Silver the following Actions need to be completed:

A.a1 Analysis Through Checklist (REPM level 1)
A.a3 Interaction Matrices (REPM level 3)
A.3.a3 Risk Assessment – selected OG1.03 (REPM level 3)

Improvement consequences
Using an official checklist when analyzing the requirements can enable a standardized way of analyzing the requirements throughout the organization and in the project in question. This also may alleviate the risk that some parts of an analysis of a certain requirement are overlooked, i.e. if the list is followed the contents of the list (template) dictates what is to be done.

Using interaction matrices dependencies, both negative and positive, could be mapped and taken into consideration. This could prove beneficial in many ways, e.g. information about problems (one requirement affecting/contradicting another), and positive effects (dependencies making for one solution for two requirements etc.).

In the case of risk assessment of selected (volatile) requirements the benefits are that one can take future potential problems into consideration, and plan for them. It is not easy to evaluate risks, but the benefit of doing a simple try can often make a lot of difference, i.e. things you did not think of come up and you specify the potential “problem requirements” and why you consider them to be volatile. This act can by itself make you aware of, and sensitive to, potential problems.

Reaching Silver will help take issues like requirements dependencies and risks into consideration, as well as standardizing and articulating the analysis process.
4.3.4.3 MPA of Requirements Management

Total number of Management Actions on each REPM level compared with the amount of Management Actions completed and Satisfied-Explained in Project Delta.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

From this chart we can ascertain that Project Delta resides on REPM level 1, i.e. Wood when it comes to the MPA of Management. Furthermore it is interesting to notice that the model lag is fairly large under this MPA with six Actions in the category Satisfied-Explained. The Actions, and the project’s motivation for excluding them, are:

M.1.1.a2 Document Usage Description
The Requirements Document can be read and used by all different groups of users due to its generality. *COMMENT ID: 20*

M.1.4.a4 User Manual Draft
The customer ordering the system opted against having a user manual at all in the project. *COMMENT ID: 21*
Chapter Four – Project Evaluation (REPM Model Validation Part II)

**M.3.a2 Backward-from traceability**
Due to the fact that it is such a small project this is not implemented. *COMMENT ID: 22*

**M.1.1.a3 Business Case**
This is done during the Requirements Engineering process but not documented in the Requirements Engineering document. *COMMENT ID: 23*

**M.1.3.a3 Environmental Models**
The project is a system located alone with no connection to other systems, which makes the use of Environmental Models unnecessary. *COMMENT ID: 24*

**M.3.a5 Forward-to traceability**
They have developed similar systems before so they base the system on earlier technologies that are familiar. *COMMENT ID: 25*

**Improvements**
In order to reach *Silver* two additional Actions have to be completed:

*M.1.2.a2 Quantitative Requirement Description (REPM level 2)*
*M.1.4.a2 Requirements Review (REPM level 3)*

(*M.1.2.1.a1 Descriptive Diagram usage (REPM level 3) – this Action is optional and does not have to be completed as a point to reach REPM level 3.)*

**Improvement consequences**
The act of quantifying requirements is a form of refinement, i.e. it gives more information about what is needed and should be done. Not quantifying requirements can have serious repercussions - if a client assumes that the uptime will be e.g. 99.995 and the best you can achieve is 98.5 there can be a problem. Specifying 98.5 and showing this to the client catches this problem early on.

Validation of requirements through formal peer-reviews can give new views to problem areas, ensure that standards and rules are followed and help further refine requirements. The positive effect of having external (peers) reviewers is that they can help to find problems the internal developers overlooked. Formal reviews are often contractually bound and this fact can instill confidence in the project from the customers point of view. Reviews can be used as an official measurement of quality.

Reaching *Silver* will help take issues like the quantification of requirements and further the validation process of the requirements.
4.3.4.4 Action Summary
Total number of Actions on each REPM level compared with the amount of Actions completed and Satisfied-Explained in Project Delta.

<table>
<thead>
<tr>
<th>REPM level</th>
<th>Total Actions</th>
<th>Completed</th>
<th>Satisfied-Explained</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

General
From this chart we can ascertain that Project Delta resides on REPM level 0, i.e. all of the Actions on REPM level 1 are not satisfied. Project Delta follows the general curve and the model lag is large, i.e. twelve Actions in total.

4.3.4.5 Summary/Discussion
Project Delta is a small size project where the customers are in close contact with the developers. The fact that the connection between the developers and the customer is so good makes it easier for them to interact and communicate. This can in turn make the elicitation of the requirements easier. To know who to talk to during the elicitation can be time and effort saving. However there is a risk that some things are taken for granted and

Figure 4.15  Project Delta - chart 4.
thereby missed. Concerning the research of stakeholders this can be such a situation, i.e. assuming to know all parties affected by the complete system may give an incomplete picture of the real situation (*COMMENT ID: 14*). However one has to take the cost of the research into consideration also.

Requirements prioritization is crucial. In most cases requirements really do have different priority. A customer stating that “all are of equal importance” often has not thought things through. If there are few requirements, e.g. five to ten, the problem may reside in the fact that the requirements are too “large” in themselves and could be broken down into several smaller ones. Prioritization would then be easier (*COMMENT ID: 15*). If a project’s resources are limited (which they always are) prioritization almost always is a good idea. In case of problems and delays due to everything from new requirements coming up to delays in development some functionality may have to be sacrificed if additional time and/or resources are not allocated. Through prioritization you can decide what parts are expendable and which are not. Requirements priorities change over time. Change can stem from a number of reasons, i.e. what people consider important, new technologies and new requirements. However when the project being developed is of a smaller size and development is over a short time period, the task of re-prioritization may seem superfluous (*COMMENT ID: 16-19*). Important is that the choice not to prioritize (and re-prioritize) may strip the developing organization from a tool, i.e. what to do when there is no more money and time, and the system is not completed. Maybe more importantly, requirements often change over time and this could have repercussions pertaining to the whole of the project. If all requirements are critical what should be sacrificed?

The requirements document is important, all results from the Requirements Engineering process are presented here and several persons read the document. Usually the persons reading the document come from different groups with different backgrounds and knowledge levels of the area of Requirements Engineering and software development in general. This is certainly true for the domain knowledge, e.g. some developers may have problems understanding domain specific information and terms. To make the document more readable and usable for all readers it is a good idea to incorporate a document usage description giving an explanation of how the requirements document should be used/read by the different groups (*COMMENT ID: 20*). In addition different groups are interested in different things when it comes to the requirements document. A developer may want to view the prototypes while a customer representative may only be interested in business cases. A usage description is basically a help text that assists users to find what they are looking for fast.

Traceability offers many benefits. Linking the requirements from the their source in other documents/people and from previous pre-studies to the relevant requirements makes it possible to validate that a specific requirements has been interpreted correctly and contains the same content and means the same thing as intended. This may also be a time saver, i.e. not having to track down the source of a particular requirement if there are questions, but being able to go directly to the source (*COMMENT ID: 22,25*). An additional example could be in the case of a requirement (or a set of requirements) being
based on a pre-study of a certain technology. If the pre-study turns up to be faulty during development it could be useful to know what requirements are effected.

To incorporate the usage of business cases in the Requirements Engineering process is a good idea for determining why the system is useful in the organization for which the system is being developed. Furthermore business cases explain what problems are to be solved. This may not include the entire business idea or organization, but rather the part of the business where the system should make a contribution (*COMMENT ID: 23*). Understanding the underlying meaning of the system makes it possible to aim at a better result in terms of developing a system that not only incorporates what the customer asks for but also what is truly needed (if the two differ).

Domain knowledge is crucial to a successful Requirements Engineering process (this is true for the whole of the system development as well). The use of system models to present and describe requirements is a way to simplify the information given. Taking consideration to the environment where the system is to be located can be a powerful tool in learning more about the domain. The method of system models can also be used to elicit new requirements from the domain. An environmental model contains not only information of possible systems that are to be interfaced with the system, but also information of the business processes that may use the system. A system is very rarely situated in an environment where it has no interaction with other components e.g. users can be considered as components that interact with the system (*COMMENT ID: 24*). They often act as a non-computerized link between systems.

The REPM model is designed to make evaluation of the Requirements Engineering process possible. Through this evaluation some ideas for improvement may become apparent. In general the MPA of *Analysis and Negotiation* is the weak link in Project Deltas Requirements Engineering process.

Studying the graphs and data for Project Delta, it seems as if the maturity of the project’s Requirements Engineering process is such that it would be the most beneficial improvement step to fulfil the *Silver* REPM level. This would ensure improvements in many areas and also represent a consistent, coherent and well thought-through Requirements Engineering process. The level of *Silver* will add useful steps for example validating the requirements through reviews, looking at the dependencies of the requirements with the help of Interaction Matrices, and further the analysis of the requirements in a structured way with the help of checklists. Therefore our improvement suggestions above are focused on achieving this level of maturity. What parts should and should not be modified/implemented is up to the organization at hand due to the fact that they are intimately familiar with their own situation.

### 4.4 Conclusion

The method for using the REPM model, i.e. the REPM evaluation checklist, proved to be an easy to understand and an effective way of conducting project evaluations. The interview subjects (project primary and in some cases others) did not seem to have any
difficulty following the structure of the project evaluation checklist (See chapter 2.3.1 and Appendix III) and its connection to the model. In some cases the questions presented in the checklist needed further clarification and exemplification. This was achieved through the REPM model and reading of the corresponding Actions. In general we feel quite confident that the method used is fairly satisfactory and adequate for the task of project evaluation.

Interpretations of the results following the diagrams, in combination with suggestions for improvement seems to be a good way of (See section 2.4) presenting the results from the project evaluations conducted. Diagrams are a good complement to the text holding our interpretations and they offer the basis for the results. Detailed results from every project evaluation, i.e. answers to every question on the checklist, were not presented as this would not serve any real purpose. The exceptions and Actions not completed are presented as these are interesting in terms of the conclusions drawn as well as for the purpose of giving suggestions for improvements. The results of the project evaluations, including our interpretations and suggestions (basically as shown above), were sent to each Project responsible for approval. This gave the Project responsible time and opportunity to comment on the presentation as well as our interpretations and suggestions.

Looking at the project evaluations as far as the REPM model is concerned the model lag was moderate in most cases, i.e. the model was general enough to be applicable to each project’s Requirements Engineering process. The model lag was four Actions for Alpha, five for Beta, four for Gamma and twelve for Delta. In the case of Project Delta twelve Actions may seem to break the trend but this is not the whole truth. Five of the Actions not completed (and deemed to fall under the category Satisfied-Explained) are dependent on each other, i.e. four are dependent on the first (See section 4.3.4.2), thus in reality the model lag effectively falls to a more moderate level of seven Actions in the category of Satisfied-Explained. On the other hand seven is the largest number in any case and the explanation for this is presented in section 4.3.4.5.

Our decision to make the Project responsible in charge of what Actions should be in the category Satisfied-Explained (See section 2.3.2.1) seems to have been correct looking at the projects evaluated above. Our misgivings in terms of misuse (See section 2.3.2.1) do not seem to be a source for concern thus far.

Based on only four project evaluations, i.e. two from each country, it is not possible to generalize when it comes to the REPM model’s applicability internationally. What we can say is that in our experience (from the project evaluations above) the REPM model and the method of using it works independent of geographical location as far as Sweden and Ireland are concerned. For a more empirically based assumption of applicability several more projects have to be evaluated.

Looking at the project evaluations above and especially our improvement suggestions we always recommend a homogenous REPM level across the three MPAs, the reason for this being mainly dependencies. If a project’s Requirements Engineering process resides on Platinum when it comes to the MPA of Requirements Elicitation but only Bronze
pertaining to the other two MPAs the whole of the process would be left staggering. In many cases Actions under one MPA are dependent on other Actions being completed under another MPA. It is like a chain, i.e. only as strong as its weakest link (See section 2.3.2). This is the main reason for not recommending a higher level for one certain MPA.

The MPA of Requirements Management is generally the one needing most improvements, i.e. the MPA with most Actions that are not completed. There are several reasons for this. Requirements Management is the largest MPA, i.e. housing the largest number of Actions, and thus there are more to be completed. Another reason may be that the Actions under this particular MPA are fairly advanced, i.e. on a higher REPM level relative to the total number of Actions. An example of this is that there are more Actions under the MPA of Requirements Management on REPM level 5 than there are under the other MPAs in total. One could argue that this mismatched situation could be attended to by splitting up the MPA of Requirements Management into two or more MPAs. This is a possible solution but we do not feel that it is a good one. The MPA of Requirements Management is the largest for a reason. All Actions general for the entire Requirements Engineering process, e.g. ongoing tasks like re-prioritization, validation and traceability are housed here. This makes this MPA the largest and the one driving the Requirements Engineering process beyond the initial tasks conducted.

What should a company do after being presented with a project evaluation like the ones above? The first step is to study the evaluation, examine the improvement suggestions and the conclusions drawn for inaccuracies and/or neglected parts. It is also important to scrutinize the Actions deemed to be under the category of Satisfied-Explained to ensure that they are put there for the right reason. After the evaluation review a plan should be constructed based on the improvement suggestions and improvement consequences. This plan should not only include what should be done but a time table so that every step (improvement) is taken in small incremental steps (See section 1.4.2). Continuous process improvement should be based on small evolutionary steps taken after careful scrutiny. Why, how and when a certain step should be taken are questions that always should precede any change being made. In addition the cost of every change should be estimated and weighed against the potential gain.

The distinction between change and improvement should be made by the ones responsible for the plan and the implementation (hopefully the same ones) of the changes stemming from the project evaluation.
5 Chapter Five – Conclusions and Discussions

The purpose of this chapter is to present the conclusions drawn from the process of developing and validating the REPM model and the project evaluation method.

This chapter is divided into four main parts; first a summary of what was done and how it is connected, second a discussion of the REPM model, third the conclusions drawn, and last there is a part suggesting future work involving the REPM model.

5.1 Summary

The main purpose of this thesis was to make a contribution to the field of Requirements Engineering. This was done by constructing a model for evaluating a project’s requirements engineering process, i.e. the Requirements Engineering Process Maturity model. The model was inspired by the authors’ first hand experience in industry, existing models like ISO 9000 and CMM, and last but not least several books and articles written in the field (See chapter 1).

After the REPM model was designed and constructed the second phase (See chapter 3) of the work was initiated – validation of the model. The model was validated through an interview done in industry with an experienced and skilled representative working with Requirements Engineering. This validation resulted in several changes and modifications to enhance and develop the model.

The third phase involved constructing a method for using the model to evaluate projects, the Project Evaluation Checklist was constructed and validated initially through peer-review (See chapter 4).

In the last phase of the thesis the REPM model (and the method for using it) was further validated through using it in industry doing “live” project evaluations, analyzing the results and developing results from each project evaluation sighted on process improvement (See chapter 4).

5.2 The REPM Model – Problem Discussion

We have identified two main things that are at the root of most problems with the REPM model in terms of design and usage; Size vs. usability and generality vs. applicability. In addition the complexity escalates as these points affect one another. Below each of the two are gone through as are some of the correlations between them.

5.2.1 Size vs. Usability

Some of the SPAs in the REPM model need to be further granulated; domain knowledge (See section 3.2.1) is an example of this. Granulation is needed for the model to house
more information about every area, i.e. every SPA is granulated further into additional SPAs and Actions that are more detailed and thus not open for the same amount of interpretation. If Actions are too abstract, i.e. high-level, chances are they can be interpreted in many ways, thus allowing an organization to deem an Action completed when in reality it is not (or only partially so) (See section 3.3).

The REPM model is designed with project evaluation in mind, but it can be used as a checklist of sorts of what could and should be done (See section 2.3). The “checklist mode” is enhanced as the model grows, i.e. there are more detailed suggestions of what to do, but the usability of the model may suffer at the same time. A main point in the REPM model development was for it to be usable without having to devote vast resources, i.e. man-hours, to complete a project evaluation. As the model grows the amount of time that has to be put into a project evaluation also grows (See section 3.3). On the other hand, the model’s usability may also grow with size, i.e. if some things are too simple (not granulated enough to be of any real value) they are likely to produce inaccurate and not as meaningful results, provided you incorporate the results produced by the model into the term usability.

The problem is where to draw the line. On the one hand we have a simpler (smaller and not so granulated) model that is fast to use but may produce lesser results, on the other we have a larger model that escalates the cost of using the model but potentially producing better results. Accuracy may be better as the model grows, but one also has to take the matter of complexity and applicability into consideration.

5.2.2 Generality vs. Applicability

A general model where much is left to interpretation (what a certain Action involves) can be applied to a wider range of projects, while a more detailed model (spelling out, in greater detail, what Actions should involve) narrows the applicability of the model to certain types of projects. Following this reasoning the model lag (See section 2.3.2.1) should grow as the model grows – keeping in mind that model lag is a measurement of the REPM models in-applicability to certain project’s Requirements Engineering process (See section 3.3). The proverb “you can not please everyone” applies here, i.e. some model lag is unavoidable (See section 4.3), but once again it is a judgment call. Is it better to have a large model with a large model lag as a result or should the model be kept smaller effectively “hiding” the model lag in the interpretation of a certain Action?

The central thing here is granularity, i.e. how low-level an Action should be. Low-level means less interpretation, but it also means a model that is applicable to fewer projects’ Requirements Engineering processes.

In the scenarios above we have two extremes. One is a general (small) REPM model with low granularity (high-level Actions), applicable to most projects but producing uncertain results and a low model lag. The other being a granulated (large) model with (low-level Actions) applicable to fewer projects, but producing more accurate results and a high level of model lag.
5.3 The REPM Model - General Conclusions

We feel that a middle road is preferable, leaning towards the first of the two extremes, i.e. less granularity. Keeping it simple is important. If the persons involved in the evaluations feel that the model is too large and complex chances are that they will not use it. In that case the better results are never produced in any case. The purpose of the REPM model has to be considered. If the purpose is to generate good results no matter what the cost of doing the evaluation there is no problem, however this is seldom the case. The cost-benefit ratio has to be acceptable, not only in terms of making money from a future improved Requirements Engineering process, but also in terms of overcoming the initial resistance. With initial resistance we mean the perception of the value of the model has to be positive, i.e. arguing for using a model and doing an evaluation for two or three hours is much easier than trying to sell the idea of a two day evaluation.

That some Actions are on a very high level, i.e. general and open to interpretation, is not necessarily a large problem. Using the model is partly a matter of conscience. Evaluations are intended to assess a particular project’s process maturity in Requirements Engineering, but not with the intention of sitting in judgment over it, but rather to evaluate its potential for improving upon the Requirements Engineering practices. Therefore, if an Action is very high-level, and the project responsible feels that there may be parts of, or interpretations of, the Action that are not covered, this should be seen as a potential for improvement where the project should at least consider the applicability of the interpretations not already covered. Hence, the model acts as much as an evaluation tool as it does a reminder of areas to consider when conducting Requirements Engineering. It is then up to the company or project to decide how much effort to put in to the area hinted at by the REPM model. It may also well be that, in this process, Actions are invented that are not present at all in the current version of the REPM model. If an organization feel the need to tailor the REPM model to suit their special needs, i.e. basically what happens when Actions are added, this would allow for low-level Actions without a high model lag (See section 2.4).

The goal is to help organizations better their Requirements Engineering process. The REPM model accomplishes this to a satisfactory extent in the case studies presented in chapter 4. The model does not offer features enabling an evaluator to elicit detailed descriptions of how a certain improvement should be carried out, nor a detailed list of what exactly is incorporated in a certain improvement. This would demand the presence of low-level Actions. What it does offer is high-level suggestions, i.e. a point by point summary comprised of Actions not completed during the Requirements Engineering process. The interpretations of these Actions and what is accredited to them decides what the steps taken should entail.

However the results produced by the model are only as good as the presentation of them. Early on in the model’s development we realized that a simple “your project resides on REPM level X” would not be adequate (See section 3.3). This prompted the use of diagrams and presenting the MPAs separately. This enables us to give a more detailed picture of the Requirements Engineering process in a structured way. Equally, if not more, important to presenting the results is the presentation of the improvement
Chapter Five – Conclusions and Discussions

suggestions for every MPA. In addition to this it is crucial to keep an eye on the Actions falling into the category Satisfied-Explained (model lag). By default Actions are to be considered completed or uncompleted, leaving only the ones not applicable to be categorized as Satisfied-explained. It may be tempting to deem Actions not applicable too often – diminishing the evaluation. We feel however that our decision to put this up to the project responsible was an accurate one, and as mentioned above the entire evaluation is to some extent a matter of conscience. The project responsible is the one with the knowledge to make the decisions, and ultimately the one affected by them (See section 4.4).

We have shown that the REPM model in combination with the Project Evaluation Checklist is a good way to evaluate the Requirement Engineering process of a project (See section 4.4). It gives a picture of the current state of the Requirements Engineering process in a project and, more importantly, how the results of the evaluation can be used for improvement. Another one of our goals was to construct a model that was usable in term of time spent vs. result. We feel that we have achieved a good balance, i.e. a project evaluation takes approximately two to three hours, and identifies real improvement opportunities.

Furthermore our second validation of the model and method was done in an international setting evaluating projects both in Sweden and in Ireland. It is difficult to compare the project evaluations done in Sweden to the ones done in Ireland, as the companies and project sizes are not compatible. To make such a comparison additional projects should be evaluated. It would have been beneficial to evaluate several more projects in an international setting, but we feel we have proven that the model and the evaluation method are applicable internationally, as far as Sweden and Ireland are concerned.

Important to remember is that the REPM models design is centered on the notion of continuous improvement. If one starts from the assumption that everything can be improved, and improvement is your goal, you need to know three things; where you stand at present, where you want to go and how to get there. The REPM model can help with two out of these three.

5.4 Future Work

The REPM model’s basic design is centered on the assumption that the initial model has potential for improvement and an increase in granularity. During this thesis we realized many things that could be done to further improve the model. Some of them are presented below in a summarized form.

- Further granulation of some of the SPAs may be beneficial (See Chapter 3).
- Development of new presentation techniques i.e. diagrams, incorporating the Optional Actions.
Chapter Five – Conclusions and Discussions

- Development of a computerized evaluation tool, i.e. computerizing the evaluation process may be beneficial in terms of access, cost and speed. The negative aspect is that the human interaction during the evaluation, i.e. clarification etc. will be lost.

- Upon the creation of a computerized tool for the REPM model (used by companies) one could gather statistics from the companies to build a database of the state-of-practice of Requirements Engineering.

- Enable companies to submit experiences from improving their Requirements Engineering process according to the REPM model. This creates a network where companies may learn from each other.

- Extend the explanations and examples of certain MPAs, SPAs and Actions. This is especially useful if the evaluation is computerized and the human interaction is lost. This can take the form of an “additional-information-button” offered to the project responsible to click on if something is unclear.

- The REPM model can be developed/adapted to act as a teaching tool of sorts, e.g. aimed at customers. This can help further the knowledge of Requirements Engineering and its importance.
6 References


References


Appendix I

Authors: Tony B. Gorschek

Kaarina Tejle

Blekinge Institute of Technology
Department of Software Engineering and Computer Science
June 2002

© Tony Gorschek and Kaarina Tejle 2002
Permission to reproduce this document and prepare derivative material for internal use is permitted provided the copyright statement is included.
Table of contents – Appendix I

Part I - Manual
Table of contents – Appendix I ........................................................................................................1
1 Introduction ......................................................................................................................................2
2 The REPM Model .........................................................................................................................3
   2.1 Introduction .........................................................................................................................3
   2.2 Structure .............................................................................................................................3
   2.3 Requirements Engineering Process Maturity .................................................................5
       REPM 1 – Initial (Wood) .................................................................................................6
       REPM 2 – Basic (Bronze) .................................................................................................6
       REPM 3 – Formulated (Silver) ........................................................................................7
       REPM 4 – Developed (Gold) .............................................................................................7
       REPM 5 – Advanced (Platinum) .........................................................................................8
3 Usage ........................................................................................................................................9
   3.1 Project Evaluation ..............................................................................................................9
       3.1.1 The Result .................................................................................................................10
       3.1.2 Satisfied-Explained Actions ..................................................................................11
       3.1.3 Result Presentation through Diagrams .................................................................13
4 References ................................................................................................................................14

Part II – REPM Model

Part III – Action Summary
1 Introduction

The area of Requirements Engineering (RE) is very interesting and often underestimated in value in the area of Software Engineering. The failure rate of IT investments is over 60%. One of the major causes for this is the lack of a complete and/or adequate requirements specification [4, 5]. The requirements specification is a direct result of the Requirements Engineering Process and it stands to reason that an inadequate specifications is a result of a RE process with a low maturity level.

The interesting thing is that much of the time spent early in the Software Development Process at RE in many cases seems unaffordable, but time spent later, in redesign and testing seems to be a point of practice in many projects. We do not agree with this paradigm, and feel that a mature RE process can help alleviate some problems that in many cases lead to budget overruns when it comes to time and resources.

It is not easy to measure the maturity of a RE process for a certain project, and subsequently it is difficult to know what is missing and what could be done to improve the process. We propose to design a model for Requirements Engineering. This model supposes to cover the area of RE, i.e. what should and/or could be done during the RE process. The model can be used to evaluate the Requirement Engineering Process Maturity (REPM) for a certain project. A spin-off effect of this is that the model basically gives the user, i.e. the project evaluator, a blueprint for how to improve the RE process.

The REPM Model is divided into three parts, Part I – a brief manual and overview of how the REPM Model is structured, Part II – the REPM Model itself, and Part III – a summary of the Actions in the model according to REPM Level. Users are recommended to read Part I before using the model.
2 The REPM Model

2.1 Introduction
The REPM model is basically a map of sorts, describing the RE process and its constituents. There are three main areas; Elicitation, Analysis and Negotiation and Management. Under each of these there are a number of sub-areas and ultimately at the bottom there are actions – describing what should be done or what should be present. Every action is mapped to a certain Requirements Engineering Process Maturity (REPM) Level (spanning from 1 to 5). If one does a cross section and takes out all actions designated Level 1 you in essence have the constituents for making a basic Requirements Specification and subsequently you have a basic RE Process, i.e. a REPM Level of 1.

2.2 Structure
The model is structured according to the following:

E Requirements Elicitation

The process of discovering the requirements for a system by communicating with the persons involved in a system is called Requirements Elicitation. The persons involved with the system are for example the customers, system users and all others who have a stake in the system and its production.

To elicit the requirements properly one needs several different areas of domain knowledge as well as knowledge over the organisational structure.

Actions

E.1 Requirements Reuse

One should not forget, when developing a new system, to reuse the requirements from other systems developed in the same application area when possible. The positive aspects of reusing requirements are that you save time and money and that you reduce risk since the requirements have already been implemented successfully before. Other areas like design and testing are also to large extent touched when it comes to the reused requirement.

Relation: IN, 1.M.D.

E.1.1 Stakeholder Identification

The stakeholder identification process is a part of the elicitation process. It means to explicitly identify all potential stakeholders. Stakeholders are people or organizations who are affected by the system and have an direct or indirect influence over the system requirements. An example of stakeholders are end-users, customers and system users. It is important to remember that the developers of the system, managers etc. also are considered as stakeholders.

Actions

E.1.1.1 Ask Executive Stakeholders

This action is basically asking the executive stakeholders (i.e. those ordering the system) who the stakeholders are.

Relation: none

E.1.1.2 Research Stakeholders

Doing your own investigation of who the different stakeholders are.

Relation: none
1 – Main Process Area (MPA)
There are three main Process areas. Each MPA has a unique identifier, e.g. E for Requirements Elicitation. This identifier’s purpose is to facilitate traceability throughout the model.

2 – Sub-process Area (SPA)
The identifier, E.1, denotes that the SPA belongs to the MPA E.

3 – Action
Actions are the final nodes in the tree-structure that the model is comprised of. Actions do not have any sub-areas/actions. An action is basically a measure that can be taken. An example can be the action E.1.a1 Ask Executive Stakeholders which is the act of asking the executive stakeholders to identify other stakeholders. An action can also be an item that should be obtained, e.g. M.1.1.a1 Summary which denotes that a summary is needed in the Requirements Document.

All Actions can be identified through the small letter “a” which precedes the Action number, e.g. in the case of M.1.1.a1 the “a” denotes that it is an Action (M.1.1.1 would signify a sub-process area).

4 – REPM
Actions are classified according to REPM Levels. The levels spans from 1 to 5.

5 – Relation
The relation denotes a dependency of sorts between actions. This generally means that an action on e.g. REPM Level 3 is dependent on that one or more actions at a lower level are fulfilled, note that this is not always the case, e.g. one of two dependent actions can be optional (See below).

Optional Actions
In some cases Actions in the REPM Model are optional, i.e. they do not have to be satisfied by a Requirements Engineering process (Figure 2.4). There are two types of Optional Actions; Optional (denoted “Opt” in the model) and Optional Groups (denoted by “OG” in the model). Opt stands for optional and this Action is voluntary, i.e. it does not have to be satisfied in order for a Requirements Engineering Process to reach a certain REPM Level. OG denotes a group of Actions of which at least one has to be satisfied in order for a project’s Requirements Engineering process to qualify for a certain REPM Level. An example of this can be seen in Figure 2.4. In this example, we have three Actions; Risk Assessment – individual, sets and selected. Each of these has an additional group identifier, i.e. OG1.01, OG1.02 and OG1.03. This denotes that all three belong to the same Optional Group (OG1) and the last number denotes the group id of each Action. For a project to reside in REPM level 3 OG1.03 has to be satisfied. If a project is to satisfy REPM level 4 at least one of OG1.01 and OG1.02 has to be satisfied, i.e. if OG1.03 is satisfied or not does not matter if REPM level 4 is the goal.
2.3 Requirements Engineering Process Maturity

Every Action is mapped to a certain Requirements Engineering Process Maturity (REPM) Level spanning from 1 to 5 (See below). The motivation behind placing an Action in a certain REPM Level is comprised of two main parts, cost and complexity. Cost denotes how much resources, e.g. man-hours and/or money, must be spent in order to satisfy an Action – the more costly the higher the level. An example can be the Action of E.1.a1 Ask Executive Stakeholders which is situated on REPM Level 1 and E.1.a2 Research Stakeholders which is situated on level 2. It is cheaper in terms of man-hours to ask the executive stakeholders, i.e. the ones ordering the system, than investigating who the stakeholders are. Complexity denotes how complex a certain Action is. An example can be the Action of E.3.a3 Technical Domain Consideration (taking the system technical operating environment into consideration) which is situated on level 1 and E.3.a1 Human Domain Consideration (taking organizational and political factors into consideration) which is situated on level 4. For a software development company it is often a necessity to investigate and take the technical aspects of the system’s operating environment into consideration, but how many take political factors, e.g. an employee who is reluctant to speak freely when the boss is present, into consideration when eliciting requirements?

In essence the REPM levels denote how advanced and mature a Requirements Engineering process for a certain project is, i.e. a higher level denotes a higher level of maturity. This does however not the same as saying that all companies can, or even should, try to get all their projects to reside on the highest level. It costs resources to reach a higher level of maturity, whether it is the Requirements Engineering process or any other part of the development process. A REPM level of three may be adequate for a certain type of project. What level a certain type of project should reside upon we leave to the organization handling the project.

The five REPM Levels are presented below.
REPM 1 – Initial (Wood)
The Actions at REPM Level 1 are what is needed to make a basic requirement specification. The Requirements Engineering process is very slim at this level and not necessarily repeatable, i.e. not something that is repeated in the same way project after project, but rather a thing of chance.
Organizations typically do not provide a stable environment for development. During a crisis, projects typically abandon planned procedures and revert to coding and testing. In addition no validation or review of the requirements takes place.

REPM 2 – Basic (Bronze)
Level 2 denotes a more structured and complete Requirements Engineering process than level 1. The goals for this level are;

A. Introduction of traceability
B. Introduction of validation of requirements
C. Introduction of a standardized structure for the documentation produced as a result of the Requirements Engineering process, i.e. the Requirements Document
D. Stakeholder identification

An organization at this level has introduced policies that ensure that requirements are specified and documented in a standardized way. This ensures repeatability inside a project, i.e. every requirement is elicited, documented and verified in the same way. The Bronze level in general denotes that an organization has devoted resources to the Requirements Engineering process as a separate entity in the Software Engineering process as a whole.
At this level the environment of the system being produced is only studied in passing, i.e. no notice is taken to the application domain or the business processes already present. Stakeholders are identified. Requirements may be insufficient in detail and/or in number, as well as incompatible with the systems operating environment.
We expect to find that Companies/organizations producing projects at this level tend to be smaller, up to 50 co-workers, and tend to be fairly young, i.e. not been in operation for more than 5 years. If we look at the domain of the organizations at this level we will probably find smaller companies involved in the development of information systems.
REPM 3 – Formulated (Silver)

Level 3 denotes a more active examination of the system environment. The goals for this level are:

A. Application domain and processes are studied and taken into consideration
B. All stakeholders are consulted
C. Dependencies, interactions and conflicts between requirements are taken into consideration
D. Requirement categorization and prioritization
E. Requirements re-prioritization
F. Peer-reviews
G. Risk assessment

The system’s environment is studied in greater detail, not only the technical aspects but also the demands coming from the application domain, as well as the business processes which the system should support. This ensures a deeper understanding of the environment in which the system is to operate, and subsequently enhances the ability to design the requirements. All stakeholder groups are consulted and reviews are conducted, i.e. peer-reviews that may or may not be contractually bound. The requirements are prioritized, and re-prioritized in case of new releases and/or new requirements. Furthermore interactions between the requirements are mapped through the use of e.g. Interaction Matrices. This alleviates the risk of conflicts. In addition to this risk assessment is conducted on selected requirements, i.e. requirements that are likely to change.

At this level no structured risk assessment is performed on sets or individual requirements. Furthermore no consideration is taken to the human domain, e.g. political and emotional factors. In addition the question of whether or not the system will add value to the organization is not raised.

We expect to find that Companies/organizations producing projects at this level tend to be moderate in size, up to 100 co-workers, and tend to be fairly seasoned, i.e. been in operation for 6 to ten years. If we look at the domain of the organizations at this level we will probably find established companies involved in the development of information systems.

REPM 4 – Developed (Gold)

Level 4 denotes a more active and mandatory examination of risks and the true value of the system seen from the organization where the system is to be implemented. The goals for this level are:

A. Human domain consideration
B. Business domain consideration
C. Advanced risk assessment
D. Advanced traceability

Consideration is taken to human domain aspects, e.g. political and emotional factors which can greatly influence requirements sources. How the system developed makes a contribution to the business at hand is also studied. This will help to drive the elicitation process and goals are set higher, i.e. more can be expected from the system on
completion. The requirements are refined through the use of scenarios at a larger scale than before, and the requirements are validated through inspections (walkthroughs). Risk assessment of the requirements (individual and/or sets) is done, e.g. risk is weighed against potential gain of a certain requirement. Traceability options are more advanced and cover documents preceding the requirements document, e.g. links pre-study documents to the relevant requirements.

This level is fairly advanced and all the major components of a well scrutinized and standardized Requirements Engineering process are present. However a planned and systematical requirements reuse structure is not present. The system architecture is not studied and may bring unwanted results in terms of unexpected interactions between subsystems.

We expect to find that Companies/organizations producing projects at this level tend to be large in size, up to 300 co-workers, and tend to be fairly seasoned, i.e. been in operation for more than ten years. If we look at the domain of the organizations at this level we will probably find established companies involved in the development of information systems as well as companies involved in critical system development.

**REPM 5 – Advanced (Platinum)**

Level 5 denotes advancement from the previous level when it comes to reuse and architectural considerations. The goals for this level are;

A. Requirements reuse  
B. Rejected requirements documentation  
C. Architectural modeling  
D. Advanced validation  
E. Advanced requirements re-prioritization

Requirements reuse is taken into consideration and reuse is done when possible. Also rejected requirements are documented as a part of the Requirements Engineering process. This documentation offers clarity (you have what should not be implemented on paper) as well as material for future reference. System model paraphrasing is used to further validate requirements as well as the creation of architectural models to map the communication between the system as a whole and the environment, but also the communication within the system, e.g. between sub-systems. The requirements are re-prioritized with regularity, independent of whether of not something happens.

We expect to find that Companies/organizations producing projects at this level tend to be large in size, up to 300 co-workers and above, and tend to be fairly seasoned, i.e. been in operation for more than ten years. If we look at the domain of the organizations at this level we will probably find established companies involved in the development of information systems as well as companies involved in critical system development with very large projects demanding vast resources.
3 Usage

The REPM model can be used for several different things. Primarily it is intended for evaluation of the maturity of the Requirements Engineering process of a certain project. The reason for it being designed towards projects and not the whole of an organization like e.g. CMM is the fact that the Requirements Engineering process varies with projects. This variation can be a result of an incomplete Requirements Engineering process, i.e. a non-repeatable one (equivalent a process at REPM level 1. But it can also be a result of the fact that different projects merit different levels of Requirements Engineering. An example of this can be an in-house project where for example risk assessment does not have to be done to the same extent as in a traditional developer-customer relationship, as the risks have already been assessed before the go-ahead was given. The usability of the REPM model does not decrease, according to our opinion, due to its project focus however. If one desires to measure the REPM level of the organization this is still possible through measuring all of the projects – thereby evaluating everything produced by the organization.

A direct spin-off effect of using the REPM model as an evaluation tool is that it produces a blueprint of the status of the Requirement Engineering process of a certain project. This can be used to see what is done, what is not, and in continuation what can be done to improve the process. With improvement we do not necessarily mean making it to the next REPM level, but rather getting to the level best suited for the project at hand. It is important to notice that a REPM level of five is not necessarily the “best” thing. All process improvement cost resources and one has to weigh the benefits of a potential improvement against the cost of achieving that improvement. Whether or not a change is to be made, or on what REPM level a certain project should reside on for that matter, is up to the people in charge. Our goal is only to give developers an easy way to assess the situation, what they do with the data is up to them.

In addition to using the REPM model for evaluation purposes it can be used as a framework of what things to do during a project. The model is not designed for this, e.g. no chronological ordering of the Actions is made, but it is conceivable that a summation of the Actions (See Appendix I - Action summary) can act as a checklist of sorts.

3.1 Project Evaluation

In Appendix III we provide a checklist to use when evaluating projects. The checklist questions follow the model and primarily its Actions. The default procedure is that one question mirrors one Action in the model. However, in certain instances more than one question is posed to ascertain if a certain Action is completed.
The basic structure of the REPM evaluation checklist is the same as for the model. It is sorted according to MPA and SPA. The Action UID denotes which Action in the REPM model the question is linked to. This so that the person(s) answering the question can look in the model for clarification if need be. In some instances a general question is asked with a MPA or SPA as a template, this to determine if any of a group of Actions is satisfied, i.e. a question like “do you do any risk assessment” is based on the SPA preceding Actions about particular risk areas. If the answer is “no” on the first question there is no need to answer questions on specific risk assessment issues. Furthermore if “no” is checked the person answering the question can (and should) clarify why, i.e. to make it possible to deem if an Action is Satisfied-Explained.

3.1.1 The Result

When the REPM evaluation checklist is completed the results are added up according to REPM level. A project has to complete all Actions at a certain REPM level to qualify for the level in question (special rules for optional Actions and Actions deemed Satisfied-Explained), i.e. for a project to reside on a REPM level of 3 all Actions at level 1, 2 and 3 have to be completed. A “perfect” example of this is:

- REPM level of Requirements Elicitation = 3
- REPM level of Analysis and Negotiation = 3
- REPM level of Requirements Management = 3

\[(3 + 3 + 3)/3 = 3\]

This gives a total REPM level of 3. If there are some completed Actions on level 4 and 5 these are disregarded in the overall REPM level (they are however present in the presentation of a project evaluation, see below).

A more realistic example we think would be when the REPM level of different MPAs differ, e.g.

- REPM level of Requirements Elicitation = 3
- REPM level of Analysis and Negotiation = 2
- REPM level of Requirements Management = 4
\[ (3+2+4)/3 = 3 \]

This also gives a REPM level of 3 in total, but the difference is that it is easy to see that the MPA of Analysis and Negotiation is at a lower level than the other two. At this stage it is possible for the organization testing its project to see in what area the process may be insufficient.

It is important to notice that the REPM levels should be homogenous, ensuring that dependencies between the different MPAs are satisfied and that there is a consistent level of maturity in the work being done, i.e. a chain is only as strong as its weakest link. An example of this can be an organization with REPM level 1 for the MPA of Requirements Elicitation and a REPM level of 5 when it comes to the MPA of Requirements Management. This would mean that advanced requirements validation is performed on requirements that are poorly elicited in the first place.

### 3.1.2 Satisfied-Explained Actions

Another factor that has to be taken into consideration when using the REPM model for project evaluation is that certain projects do not merit the use of all Actions. This can be due to many reasons, e.g. internal projects (in-house development) may not merit the use of e.g. a detailed risk analysis or the need to research stakeholders etc. A company devoted to carrying out projects in special environments, i.e. a variant of the traditional customer-developer environment, may deem certain Actions unnecessary. An example can be a company where the developer and the customer both have specialized on a certain domain and “speak the same language”. The need for extended clarification and validation of requirements may not be needed, e.g. the construction of prototypes can be omitted. This is not the same as optional Actions however. Optional Actions are Actions that we consider not to be vital to the Requirements Engineering process, but may be a good complement. All Actions not marked optional are considered vital.

In order to take all of these factors into consideration we have chosen to add a new term to the REPM model, Satisfied-Explained. This expression denotes an Action that is not completed - but the organization doing the evaluation deems the Action not applicable to their project. Satisfied-Explained is used to counter the effect of misleading results when evaluating a project.

An example of this can be a company not using the Action of E.4.a3 In-house Scenario Creation. Let us say that the developing company works in, and is specialized on, one domain. All of the customers are also specialized in the same domain. In addition the developing company sells only one system, highly specialized to suit the domain and it is only adapted to some extent for every customer. The developing company deems the Action E.4.a3 In-house Scenario Creation to be unnecessary due to the fact that they have much experience of adapting and selling the system, and do not need to construct in-house scenarios to clarify and validate the requirements at hand. When doing an evaluation of the project in question the REPM level would not even reach level one (REPM 1) due to the fact that E.4.a3 In-house Scenario Creation (which is on level 1) was not completed. The rest of the Actions up to REPM level 5 could be fully completed but the REPM level of the project would be zero anyway. This is misleading and does not give an accurate picture of the REPM level of the project. For this reason Actions are considered satisfied both when actually completed and when they are deemed to be
inapplicable and/or irrelevant to the project at hand, in this case the Action is said to be Satisfied-Explained (not completed).

The organization doing the evaluation makes the distinction when an Action is to be considered Satisfied-Explained. Important to notice is that an Action is not deemed Satisfied-Explained for reasons like lack of time, lack of money, lack of know-how or just “did not think of it”. The lack of time and money is fairly self explanatory, as is “did not think of it”. Know-how however refers to an instance where the competence to implement an Action is not present in the organization. Know-how can also refer to the fact that a certain Action’s benefit (the positive effect of conducting a certain task) is not obvious. It is easy to dismiss Actions (put them into the category of Satisfied-Explained) and this is a substantial risk. However this is up to the person(s) evaluating the project.

The use of Satisfied-Explained Actions, i.e. if an organization deems that one or more Actions in the REPM model falls into this category, is subjective. The one(s) doing the project evaluation make the distinction. An Action can thus after an evaluation have three states:

1. **Completed** - **The Actions is done.**

2. **Not Completed** - **The Action is not done for any number of reasons.**

3. **Satisfied-Explained** - **The Action is not completed or completed partially, but the main thing is that the Action as formulated in the REPM model is not applicable to the Requirement Engineering process of the organization evaluating the project.**
3.1.3 Result Presentation through Diagrams

The use of writing out the REPM level of a certain project in numbers is a fairly crude way of measuring the Requirements Engineering process. It can be a good overall indicator but there are better ways of presenting the results of a project evaluation using the REPM model. In Figure 2.6 an example of this is given. There are three things presented; the total number of Actions at each REPM level, completed Actions and Actions that fall under the category of Satisfied-Explained. In this example REPM level 1 is not achieved (only 8 out of 10 Actions are completed). If we look at the Satisfied-Explained graph line the two remaining (not completed) Actions are Satisfied-Explained – thus a total of 10 Actions are satisfied rendering the project on REPM level 1. REPM level 2 is not achieved, not even when taking the Satisfied-Explained Actions into consideration. In the graph the line denoting the Satisfied-Explained Actions is special. It really denotes completed Actions plus the Satisfied-Explained Actions. This is to enhance the readability and the usability of the graph. A summary of the Actions completed and Satisfied-Explained is the real measurement of a companies Requirements Engineering process maturity level. It is easy to differentiate between the two (completed and Satisfied-Explained) however as the line denoting the completed Actions is also present. For the organization doing the evaluation the area between Total Actions-line and the Satisfied-Explained-line is the most interesting. It represents the space for possible improvement of the Requirements Engineering process. The area between the Completed-line and the Satisfied-Explained-line represents the level of inapplicability of the model to a certain projects Requirements Engineering process (called \textit{model lag}).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Figure2_6.png}
\caption{Example of Total/Completed/Satisfied-Explained Actions.}
\end{figure}
4 References


E Requirements Elicitation

The process of discovering the requirements for a system by communicating with the persons involved in a system is called Requirements Elicitation. The persons involved with the system are for example the customers, system users and all others who have a stake in the system and its production.

To elicit the requirements properly one needs several different areas of domain knowledge as well as knowledge over the organizational structure.

Actions

**E.a1 Requirements Reuse**

One should not forget, when developing a new system, to reuse the requirements from other systems developed in the same application area when possible. The positive aspects of reusing requirements are that you save time and money and that you reduce risk since the requirements have already been implemented successfully before. Other areas like design and testing are also to large extent finished when it comes to the reused requirement.

Relation: M.1, M.3

---

E.1 Stakeholder Identification

The stakeholder identification process is a part of the elicitation process. It means to explicitly identify all potential stakeholders. Stakeholders are people or organizations who are affected by the system and have a direct or indirect influence over the system requirements. An example of stakeholders are end-users, customers and system users. It is important to remember that the developers of the system, managers etc. also are considered as stakeholders.

Actions

**E.1.a1 Ask Executive Stakeholders**

This action consists of asking the executive stakeholders (the ones ordering the system) who the stakeholders are.

Relation: none

**E.1.a2 Research Stakeholders**

Doing your own investigation of who the different stakeholders are.

Relation: none
E.2 Stakeholder Consulting

Another area of the elicitation process is to consult the stakeholders. This is of course directly connected to the process of identifying the stakeholders. After identifying the stakeholders the consulting process is a rather natural step forward to elicit the requirements.

Actions

- **Executive Stakeholders**
  
  These are the stakeholders that have the authority when it comes to the order/purchase of the system being developed. Typically one or two persons represent a group of executives that have taken the decision to order/purchase the system. The representatives are the ones that normally contact you and with whom you speak during the day-to-day work.

  **Relation:** E1

- **General Stakeholders**
  
  General Stakeholders is a much larger group of people. The group is comprised of all parties having a stake in the system that are associated with the customer or the world surrounding the system. Basically all stakeholders are represented here (including Executive Stakeholders), the only exception being stakeholders represented in the developing organization/organizations. End-users are a large group in the General Stakeholder group.

  **Relation:** E1

- **In-house Stakeholders**
  
  The stakeholders grouped here are the ones involved in the development/management of the system within the developing company/companies, e.g. programmers, designers, managers, legal, marketing and so on. This is a group that has a very high stake in the development and management of the system, but they are often overlooked as being stakeholders due to the fact that they are part of the development team.

  **Relation:** E1
E.3 Domain Knowledge

Domain knowledge is the general knowledge of all the different aspects and viewpoints of the system. This area is divided into several sub-areas depending on viewpoint.

Actions

E.3.a1 Human Domain Consideration

When talking about the human domain knowledge area one must consider the influence that comes from for example political and organizational factors. These factors can to a very high degree influence the requirements sources. An example of this can be an stakeholder who is reluctant to give you accurate information due to the fact that she feels pressure from the organization that employs her.

Relation: E.2

E.3.a2 System Domain Consideration

To be sensitive to the system domain knowledge leads to regarding the requirements coming from the application domain of the system. The system domain should be studied under the elicitation process. Examples of this can be in a banking system there may be accounting regulations that have to be followed. Basically look for domain constraints.

Relation: E.2

E.3.a3 Technical Domain Consideration

Technical domain knowledge involves the knowledge of the system’s operating environment. The system’s operating environment consists of all hardware and other software systems that will interact with the system being developed. This includes other systems present and third-party products like database systems.

Relation: E.2

E.3.a4 Business Domain Consideration

General information about the business concerns - how the system will make a contribution to the organization is called the business domain. The knowledge of this area will help you to drive the elicitation process forward when having higher goals to strive for.

Relation: E.2

E.3.a5 Operational Domain Consideration

When developing a computer based system the objective is often to support other business processes. These business processes might be something like systems producing customer reports or technical activities such as navigating an aircraft.

When analyzing, understanding and documenting this area one completes the domain knowledge area and drives forward the elicitation process.

Relation: E.2, E.4
E.4 Scenario Elicitation

Using the interaction sessions, known as scenarios, helps the developers see what kind of information is needed and its presentation - all from the users perspective. Basically what the customers expect, need and how they want it.

Actions

E.4.a1 Scenario Elicitation - Executive Stakeholders  REPM 2
To elicit scenarios from Executive Stakeholders and assume that they know what the end-user scenarios look like.
Relation: E.2.a1

E.4.a2 Scenario Elicitation - General Stakeholders  REPM 4
To elicit scenarios from General Stakeholders, e.g. end-users (often the largest group), system managers/administrators - in short all parties affected by the system.
Relation: E.2.a2

E.4.a3 In-house Scenario Creation  REPM 1
To make scenarios from the developer(s) experiences and the gathered information. This is without directly eliciting the scenarios from stakeholders. The scenarios here are based on what the developer(s) think.
Relation: E.2
A Requirements Analysis and Negotiation

When the initial set of requirements has been elicited the analyzing part begins. You should analyze the requirements for conflicts, overlaps, omissions and inconsistencies.

After the analysis the information received should be reviewed by the different stakeholders and through negotiation a set of requirements should be decided up on. Conflicts must be resolved and the requirements should be prioritized.

Actions

A.a1 Analysis Through Checklists
Checklists are useful and fairly simple tools that can help you in many areas, of which the analysis is one. Basically it is a list with steps that are more or less mandatory to go through. This way it is easy and clear what should be done for each and every requirement and you do not miss any steps during the analysis.

Relation: M.1.1

A.a2 Requirements Classification
It is beneficial to classify requirements during the RE process. The two main reasons for this is traceability and identifying related requirements. Dividing the requirements into classes or groups makes it easier to see which classes/groups are affected if e.g. a change in one requirement takes place. Furthermore you can use a standard classification policy to avoid missing requirements. Let us say that you normally have at least five requirements under every class in a normal case, if one is empty on a project chances are that you have missed something.

Relation: M.2.a2

A.a3 Interaction Matrices
Interaction between requirements are often many and not always plain to see. The use of matrices makes it easier to discover and document dependencies, conflicts and take problems and benefits into consideration. It is also possible to observe independent requirements. Interaction matrices can also be useful during the negotiation process as they can show which requirements can cause problems or be difficult to implement due to the interaction with other requirements.

Relation: M.3

A.a4 Ambiguous Requirements refinement
Requirements that are not clear can be clarified by documenting information about what they do not cover. Important to notice is that this should be done when the developers feel that it is appropriate. An example can be a printing function in the system. If a stakeholder assumes that it can print to both paper and file - and the developer knows this but the only thing being implemented is the paper function one could specify that the "print to file" option will not be included in the requirement. This is a rather over simplified example.

Relation: none
A.1 System Boundaries Definition

When starting the Analysis and Negotiation process one should define the system boundaries. This could be done by assessing the initial set of requirements. You should effectively decide which requirement lie in the scope of the system and which are outside.

Actions

A.1.a1 Boundary definition through categorization

To evaluate if a requirement is within the boundary of the system (the computer based system you are developing) you can categorize (in association with the Executive and General Stakeholders) the requirements into different groups, preferably System Requirements (inside the system scope), Requirements for the operational processes associated with the system and Requirements clearly outside the scope.

Operational processes requirements are defined by the fact that they are in need of information that lies outside the scope of the designed database of the system and subsequently in need of human decision making.

Relation: E.2.a1, E.2.a2, A.a2
A.2 Requirements Prioritization

During the Analysis and Negotiation one should divide time to go through the requirements with the intention to prioritize them. The priority should be set to reflect the importance the requirement has to the stakeholders and the overall success of the system.

Actions

A.2.a1 Prioritizing Requirements

It is imperative that you have a clear and complete picture of the requirements before this step. Putting the requirements into a fairly small number of priority classifications is preferable to having many (10 vs. 100), e.g., critical, important, useful and desirable. It is important to realize that different stakeholders may be partial to certain requirements and setting higher priority to them. Logical and Economical arguments must be presented for each and every requirement.

Relation: A.1.a1, A.3, E.2.a1, E.2.a2, E.2.a3

A.2.1 Requirements Re-prioritization

Requirements priority change over time. Change can stem from any number of reasons, of which the most common are covered by the Actions below. It is important to realize this change and to re-prioritize requirements as needed.

Actions

A.2.1.a1 Re-prioritization - New Requirements

When a new requirement is introduced there is a need to give it a priority and to take the requirement's impact on other requirements into consideration. It may be a simple matter of just inserting a new requirement and adding additional resources to the project to compensate. On the other hand there may be a need for re-prioritizing all of the requirements due to resource management and/or the fact that a new requirement may change the importance of other requirements.

Relation: A.2.a1

A.2.1.a2 Re-prioritization - New Release

New releases can give rise to changes in requirements’ priority. New release is a broad term and means everything from a new release of the system being developed, new releases from third-party products to new releases of rules and regulations governing the domain of the system.

Relation: A.2.a1, E.3.a2, E.3.a3, E.3.a5

A.2.1.a3 Re-prioritization with Regularity

During a project's life time requirements' priority often change as a rule rather than as an exception. The change does not have to be linked to a certain clear factor, like new releases or new requirements, but may come as a result of changes over time. These changes to environment and human domain factors are hard to spot and require that requirements be re-prioritized regularly. This Action denotes the fact that re-prioritization is done regularly as a point of practice and is not as such not dependent on a change being noticed.

Relation: A.2.a1, E.3
A.2.1.a4  
**Re-prioritization due to Change**  
This Action denotes re-prioritization due to all changes (except new requirements and new releases covered in previous Actions). The key point being that a re-prioritization is done when a change is noticed, not on a regular basis as a point of practice (covered in A.2.1.a3). Changes may stem from human domain factors such as changes in the organization (both in the developing company and customer company).

**Relation:** A.2.a1, E.3

---

### A.3 Requirements Risks

After eliciting the requirements and starting the analysis a risk analysis should be carried out on individual and sets of requirements. This is an excellent way of evaluating possible problems that may arise in the implementation of the requirements. You should also assess the probabilities of these problems arising and the effects of the problems if they do arise. Important to notice is that risk assessment of individual, sets and selected requirements are not necessarily mutually exclusive but may rather complement each other, e.g. the risk for a set of requirements may differ from the risk of the individual requirements making up the set in question.

#### Actions

**A.3.a1 Risk Assessment - individual**  
Each individual requirement is assessed according to certain types of risk.

The different types of risk could be: performance risk, safety and security risk, process risk, implementation technology risk, database risk, schedule risk, external risk and stability risk.

**Relation:** A.a1

**A.3.a2 Risk Assessment - sets**  
Sets of requirements, e.g. predefined classes of requirements (e.g. done in A.a2 Requirements Classification), can be assessed as a group/set. The assessment should be done according to certain types of risks, an example of these was covered in A.3.a1.

**Relation:** A.a1, A.a2

**A.3.a3 Risk Assessment - selected**  
Certain selected requirements are assessed due to the fact that they are considered within areas of acknowledged risk, such as requirements subject to frequent change.

**Relation:** A.a1, A.a3, M.4.a1
M Requirements Management

The management process encapsulates all of the Requirements Engineering areas. Management in RE can be compared to Project Management in projects. Documentation, Traceability policies and Change Policies are all examples of areas covered in Requirements Management.

**Actions**

**M.a1 Requirements Origin Specification**  REPM 2

It is important to have a clearly documented specification which links the natural language stakeholder requirement to the more detailed models specifying the system. This makes it easier to crosscheck models and associated requirements and reduces the chance for specification drift (the developer specifying the requirements can lose sight of what the system stakeholders really need/want).

**Relation:** M.1.1, M.3, M.1.2.a1

**M.a2 Global System Requirements Identification**  REPM 3

The Global System Requirements are requirements that are true for the system as a whole, not any individual system part or subsystem.

The information gathered in the areas of E.2.a2 System Domain Consideration, E.2.a3 Technical Domain Consideration and E.2.a4 Business Domain Consideration comprise the foundation for the data needed for identification of Global System Requirements.

**Relation:** E.3.a2, E.3.a3, E.3.a4

**M.a3 Rejected Requirements Documentation**  REPM 5

It is important to document all of the requirements, not only the ones to be implemented. Some requirements are dropped or rejected for a wide variety of reasons, e.g. cost and time considerations. They are to be documented. This documentation offers clarity (you have what shouldn’t be implemented on paper) as well as material for future reference.

**Relation:** M.1.1, M.2.a2, M.3

**M.1 Requirements Document**

The requirements document is a document that communicates the requirements to the customers, system users, managers and system developers. It might be of interest to show the document to all other stakeholders, but that is something that is voluntary.

The Requirements Document is basically a gathering of all the documentation produced during the RE process.

**Actions**

**M.1.a1 Record Requirements Rationale**  REPM 3

The rationale is the basis for the requirement. Information about why the requirement was specified in the first place and what function it has should be specified. This specification should be made at an early stage so that the initial rationale is documented.

**Relation:** M.1.2.a1, M.3
Standardized Document Structure

All the documents written during the requirements engineering process should be of a set standard structure. The specific structure can be directed by a company standard and should be checked as a part of the document quality assurance process.

The Actions below are parts that should be present in a Requirements document.

**Actions**

M.1.1.a1 *Document Summary* **REPM** 1
The Requirements Document should have a summary. The summary is a way to get a quick overview of the entire document without having to read it all. A reader can use the summary to quickly decide what parts he/she needs to read further.

Relation: none

M.1.1.a2 *Document Usage Description* **REPM** 2
This section is included in the beginning of the Requirements Document and gives an explanation of how the Requirements Document should be used/read. Different user groups are taken into consideration and the section should be divided into different parts addressing each user group. Each group should receive the information of how they should use the document for optimal benefit. A section about what is needed of the reader (expert-knowledge etc.) should also be present.

Relation: E.1

M.1.1.a3 *Business Case* **REPM** 3
The Business Case is basically an explanation of why the system is needed in the business at hand, what the purpose is, what problems are solved and what opportunities are taken advantage of through the system. This does not necessarily include a thorough description of the world/organization surrounding the system but rather the business case where the system is a part.

Relation: E.3.a4

M.1.1.a4 *Term Definition* **REPM** 1
This is a very important part of the Requirements Document. There are many types of users of the document and they often have different backgrounds. A Term Definition part (i.e., a data dictionary) will give everybody an equal chance to understand the vocabulary used in the document. Also the Term Definition can eliminate some ambiguity when it comes to the interpretation of the document and certain terms.

Relation: none
**M.1.2 Describing Requirements**

The natural language used to describe requirements should be concise, understandable and unambiguous. The requirements need to be written in a manner that will help all the readers to immediately understand the requirements meaning and placement.

**Actions**

**M.1.2.a1 Requirements Description Template**

The use of a template as a way to organize the description of requirements makes for a standardized specification. If a certain way of describing requirements (and what information should be present) is used at all times the reader will be familiar with the way the information is written and can more efficiently absorb the contents.

*Relation*: none

**M.1.2.a2 Quantitative Requirements Description**

This mainly applies to non-functional requirements such as e.g. user load, hardware utilization and uptime. It is beneficial to quantify (put a figure on) these requirements at an early stage so that they can be taken into consideration when specifying the other requirements e.g. avoiding conflicts between them.

*Relation*: E.3.a3

**M.1.2.a3 Unambiguous Requirement Description**

Each Requirement should be described using clear and unambiguous natural language. As far as possible a description should not be open to interpretation but rather be clear to a reader.

Furthermore it is important that every requirement be specified separately, i.e. several requirements should not be described in one text body.

*Relation*: none

**M.1.2.1 Descriptive Complements**

To best describe a requirement one might have to use other sorts of notation to make the descriptions. This could include mathematical formulas, specialized notation, decision tables or programming language. These complements should always be present if there is a need for a more precise explanation of the requirement.

**Actions**

**M.1.2.1.a1 Descriptive Diagrams usage**

The use of diagrams is another way of structuring the information in a readable way. Diagrams can be a good complement in the Requirements Document. It can visually describe several things like summation of numeric information or sequences of events and activities, e.g. WBS, Gantt, Interaction diagrams and Architectural diagrams.

*Relation*: none
M.1.3 System Modeling

Another way of simplifying the presentation and description of requirements is to use system modeling. A system model can also help to put the requirements in their proper environment and define new requirements. There are several different types of models that can be used, e.g. Prototypes, System models, Environmental models and Architectural models.

**Actions**

**M.1.3.a1 Prototyping** REPM 2

A prototype is a demonstration system that may show what facilities the system can provide. This can be successfully used in two ways first as a way of refining poorly understood requirements and at a second stage to verify that the requirements are adequately specified and/or correctly understood. Important to notice is that all of the stakeholders (i.e. the customer) understands that a prototype is only a cosmetic piece of software and not a product that is complete in any way, this to avoid misunderstandings.

*Relation:* E.2.a3, E.4, M.2.a2

**M.1.3.a2 System Models** REPM 3

System models are models of system specification information. Several models should be made of different parts of the system to simplify the overview. Examples of system models are data-processing models, composition models, classification models, stimulus-response model and process model.

*Relation:* E.2.a3, E.4, M.2.a2

**M.1.3.a3 Environmental Models** REPM 4

The Environment model is a model of the systems environment. The model includes information of other automated systems that are interfaced with the system being developed, as well as business processes that may use the system. In short information about what lies outside the system.

*Relation:* E.3.a2, E.3.a3, E.3.a4, M.2.a2

**M.1.3.a4 Architectural Models** REPM 5

Architectural models are overviews of sorts depicting the entire system, the sub-systems and how they are linked. Communication between subsystems is a crucial part of the architectural description.

*Relation:* M.2.a2
**M.1.4 Requirements Validation**

After producing the requirements document the requirements should be formally validated. This process is concerned with checking the requirements for omissions, conflicts and ambiguity. It is also the process to confirming that the right requirements are followed and that the requirements coincide with what the stakeholders have agreed upon.

A large part of requirements validation is concerned with the task of validating that standards are followed.

**Actions**

**M.1.4.a1 Requirements Inspection**

An inspection is a technique for the detection of errors, violation of development standards and other problems. It is basically a walkthrough. A Requirements Engineer leads the involved through a segment of documentation which are commented upon and then analyzed by the author. The inspection can be a peer-inspection (conducted by people not previously involved in the RE process) or not.

Relation: E.2.a3, M.1.1, M.1.3

**M.1.4.a2 Requirements Review**

Reviews are always conducted by peers (independent from the development at hand) and they are often contractually bound, e.g. at certain stages (milestones) you decide to have a review to ensure that standards are followed, errors are not present and so on.

Relation: E.2.a3, M.1.1, M.1.3

**M.1.4.a3 System Model Paraphrasing**

This is a sort of reverse engineering. Use the system models, diagrams and so on and convert the requirements into natural language. This to let the stakeholders (primarily the General Stakeholders) understand and comment on the requirements.

Relation: M.1.2.1, M.1.3.a2

**M.1.4.a4 User Manual Draft**

Using the requirements document as a system specification rather than a blueprint and write an initial user manual draft. This can unravel problems and omissions in the requirements specification.

Relation: none

**M.1.4.a5 Requirements Test Cases**

For each requirement there should be at least one proposed test case. This to check for incompleteness and ambiguity – in short to test if the requirement in question is met and implemented in a correct and adequate way in the system.

Relation: M.1.3.a1
M.2 CARE Tool Utilization

With CARE Tool we mean all Computer Aided Requirements Engineering tools, that can help the analyzing and negotiation. This can include graphic tools, text editor programs or communication programs. When encouraging the use of electronic systems to exchange information about the requirements you can make communication easier and faster. You have to observe that the use of CARE tools does not by itself solve communication problems or errors.

Actions

M.2.a1 Information Interchange Through CARE  
The use of Computer Aided tools in the RE process can be a powerful tool to improve and facilitate communication between stakeholders. Examples of such tools can be everything from simple e-mail communication to video conferencing systems that allow more frequent meetings between all types of stakeholders with minimizing the costs of such meetings.

Relation: none

M.2.a2 Information handling Through CARE  
The use of Computer Aided tools in the RE process can be a powerful tool to improve and facilitate information handling, e.g. using databases for storing information and modeling tools.

Relation: none
M.3 Traceability Policies

To define what traceability information and how the information should be represented is something that should be stated in the traceability policies and is a part of the management process. Traceability information is information which allows you to find relationships and dependencies between requirements. The relationships can also be defined between the requirements and the system design, components and documentation. In addition information about author/creator of the requirement, and the one responsible for it should be available.

It is important to note that the area of traceability is not covered in detail in this version of the REPM Model - parts like the need for a traceability manual and so on.

Actions

M.3.a1 Requirements Identification

Every requirement should have a unique identifier. All information pertaining to a certain requirement should be tagged with this identifier.

Relation: M.2.a2, M.1.2.a1

M.3.a2 Backward-from traceability

Links requirement to their source in other documents or people.

Relation: M.3.a1, E.4, M.a1, M.1.a1, M.1.3, E.a1

M.3.a3 Forward-from traceability

Links requirements to the design and indirectly implementation components.

Relation: M.3.a1

M.3.a4 Backward-to traceability

Links design and implementation components back to requirements.

Relation: M.3.a1

M.3.a5 Forward-to traceability

Links other document (which may have preceded the requirements document, e.g. pre-study documents) to the relevant requirements.

Relation: M.3.a1

M.3.a6 Version traceability

Handles the different versions and variations of a specific requirement and links them together.

Relation: M.3.a1
M.4 Requirements Change Policies

A requirement change policy should consist of the information of how to manage a change of a requirement, how the change should be proposed, analyzed and reviewed. All this to simplify and effectively change the requirements in the way the stakeholder or other persons involved wants them changed.

Actions

M.4.a1 Volatile Requirements Identification

Volatile Requirements are the requirements that are likely to change during the RE process. By identifying these it is possible to foresee and anticipate change and possible problems beforehand.

Relation: A.a2, A.a3, E.3

M.5 Documentation Deliverables

Before, during and after system development documentation is produced. This documentation can be everything from system models, diagrams, design documents, management documents to user manuals. These are deliverables just like system modules and should be present in the requirements document specified just like a system requirement.

I.e. if you are to deliver certain documentation the requirements that this documentation should satisfy must be specified along with what documents should be present.

Actions

M.5.a1 User documentation

This group is comprised of all documents used by the users of the system in question, e.g. user manuals, user dictionaries and so on.

Relation: M.1.1, M.1.2

M.5.a2 System documentation

This group covers all system documentation from presudy to complete system design with all pertaining documents, e.g. Design documentation, technical specifications, use case diagrams and so on.

Relation: M.1.1, M.1.2

M.5.a3 Management documentation

This group contains all management documentation to handle the finished system for all kind of upgrades or administrative actions, e.g. how to maintain the system, run diagnostics and optimize the system.

Relation: M.1.1, M.1.2
## Action summary

### Action REPM Level 1

<table>
<thead>
<tr>
<th>Name id</th>
<th>Action Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Elicitation</strong></td>
<td></td>
</tr>
<tr>
<td>E.1.a1</td>
<td>Ask Executive Stakeholders</td>
</tr>
<tr>
<td>E.3.a3</td>
<td>Technical Domain Consideration</td>
</tr>
<tr>
<td>E.2.a1</td>
<td>Executive Stakeholders</td>
</tr>
<tr>
<td>E.4.a3</td>
<td>In-house Scenario Creation</td>
</tr>
<tr>
<td><strong>Analysis and Negotiation</strong></td>
<td></td>
</tr>
<tr>
<td>A.a1</td>
<td>Analysis Through Checklists</td>
</tr>
<tr>
<td><strong>Management</strong></td>
<td></td>
</tr>
<tr>
<td>M.1.1.a1</td>
<td>Document Summary</td>
</tr>
<tr>
<td>M.1.1.a4</td>
<td>Term Definition</td>
</tr>
<tr>
<td>M.1.2.a3</td>
<td>Unambiguous Requirement Description</td>
</tr>
<tr>
<td>M.2.a1</td>
<td>Information Interchange Through CARE</td>
</tr>
<tr>
<td>M.2.a2</td>
<td>Information handling Through CARE</td>
</tr>
</tbody>
</table>

### Action REPM Level 2

<table>
<thead>
<tr>
<th>Name id</th>
<th>Action Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Elicitation</strong></td>
<td></td>
</tr>
<tr>
<td>E.1.a2</td>
<td>Research Stakeholders</td>
</tr>
<tr>
<td>E.2.a3</td>
<td>In-house Stakeholders</td>
</tr>
<tr>
<td>E.4.a1</td>
<td>Scenario Elicitation - Executive Stakeholders</td>
</tr>
<tr>
<td><strong>Analysis and Negotiation</strong></td>
<td></td>
</tr>
<tr>
<td>A.a2</td>
<td>Requirements Classification</td>
</tr>
<tr>
<td><strong>Management</strong></td>
<td></td>
</tr>
<tr>
<td>M.a1</td>
<td>Requirements Origin Specification</td>
</tr>
<tr>
<td>M.1.1.a2</td>
<td>Document Usage Description</td>
</tr>
<tr>
<td>M.1.2.a1</td>
<td>Requirements Description Template</td>
</tr>
<tr>
<td>M.1.2.a2</td>
<td>Quantitative Requirements Description</td>
</tr>
<tr>
<td>M.1.3.a1</td>
<td>Prototyping</td>
</tr>
<tr>
<td>M.1.4.a4</td>
<td>User Manual Draft</td>
</tr>
<tr>
<td>M.1.4.a5</td>
<td>Requirements Test Cases</td>
</tr>
<tr>
<td>M.3.a1</td>
<td>Requirements Identification</td>
</tr>
<tr>
<td>M.3.a2</td>
<td>Backward-from traceability</td>
</tr>
<tr>
<td>M.3.a4</td>
<td>Backward-to traceability</td>
</tr>
</tbody>
</table>
### Action REPM Level 3

<table>
<thead>
<tr>
<th>Name id</th>
<th>Action Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Elicitation</strong></td>
<td></td>
</tr>
<tr>
<td>E.3.a2</td>
<td>System Domain Consideration</td>
</tr>
<tr>
<td>E.3.a5</td>
<td>Operational Domain Consideration</td>
</tr>
<tr>
<td>E.2.a2</td>
<td>General Stakeholders</td>
</tr>
<tr>
<td><strong>Analysis and Negotiation</strong></td>
<td>A.a3</td>
</tr>
<tr>
<td>A.1.a1</td>
<td>Boundary definition through categorization</td>
</tr>
<tr>
<td>A.2.a1</td>
<td>Prioritizing Requirements</td>
</tr>
<tr>
<td>A.2.1.a1</td>
<td>Re-prioritization – New Requirements</td>
</tr>
<tr>
<td>A.2.1.a2</td>
<td>Re-prioritization – New Releases</td>
</tr>
<tr>
<td>A.3.a3</td>
<td>Risk Assessment – selected OG1.03</td>
</tr>
</tbody>
</table>

### Management

<table>
<thead>
<tr>
<th>Name id</th>
<th>Action Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>M.a2</td>
<td>Global System Requirements Identification</td>
</tr>
<tr>
<td>M.1.a1</td>
<td>Record Requirements Rationale</td>
</tr>
<tr>
<td>M.1.1.a3</td>
<td>Business Case</td>
</tr>
<tr>
<td>M.1.2.1.a1</td>
<td>Descriptive Diagrams usage - Opt</td>
</tr>
<tr>
<td>M.1.3.a2</td>
<td>System Models</td>
</tr>
<tr>
<td>M.1.4.a2</td>
<td>Requirements Review</td>
</tr>
<tr>
<td>M.3.a3</td>
<td>Forward-from traceability</td>
</tr>
<tr>
<td>M.4.a1</td>
<td>Volatile Requirements Identification</td>
</tr>
<tr>
<td>M.5.a1</td>
<td>User documentation</td>
</tr>
<tr>
<td>M.5.a2</td>
<td>System documentation</td>
</tr>
</tbody>
</table>

### Action REPM Level 4

<table>
<thead>
<tr>
<th>Name id</th>
<th>Action Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Elicitation</strong></td>
<td></td>
</tr>
<tr>
<td>E.3.a1</td>
<td>Human Domain Consideration</td>
</tr>
<tr>
<td>E.3.a4</td>
<td>Business Domain Consideration</td>
</tr>
<tr>
<td>E.4.a2</td>
<td>Scenario Elicitation - General Stakeholders</td>
</tr>
<tr>
<td><strong>Analysis and Negotiation</strong></td>
<td>A.a4</td>
</tr>
<tr>
<td>A.2.1.a4</td>
<td>Re-prioritization due to Change</td>
</tr>
<tr>
<td>A.3.a1</td>
<td>Risk Assessment – individual - OG1.01</td>
</tr>
<tr>
<td>A.3.a2</td>
<td>Risk Assessment - sets - OG1.02</td>
</tr>
<tr>
<td><strong>Management</strong></td>
<td>M.1.3.a3</td>
</tr>
<tr>
<td>M.1.4.a1</td>
<td>Requirements Inspection</td>
</tr>
<tr>
<td>M.3.a5</td>
<td>Forward-to traceability</td>
</tr>
<tr>
<td>M.5.a3</td>
<td>Management documentation</td>
</tr>
</tbody>
</table>
## Action REPM Level 5

<table>
<thead>
<tr>
<th>Name id</th>
<th>Action Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Elicitation</strong></td>
<td></td>
</tr>
<tr>
<td>E.a1</td>
<td>Requirements Reuse</td>
</tr>
<tr>
<td><strong>Analysis and Negotiation</strong></td>
<td></td>
</tr>
<tr>
<td>A.2.1.a3</td>
<td>Re-prioritization with Regularity</td>
</tr>
<tr>
<td><strong>Management</strong></td>
<td></td>
</tr>
<tr>
<td>M.a3</td>
<td>Rejected Requirements Documentation</td>
</tr>
<tr>
<td>M.1.3.a4</td>
<td>Architectural Models</td>
</tr>
<tr>
<td>M.1.4.a3</td>
<td>System Model Paraphrasing</td>
</tr>
<tr>
<td>M.3.a6</td>
<td>Version traceability</td>
</tr>
</tbody>
</table>
Requirements Engineering Process Maturity (REPM) Model

The questions are divided into three parts, introductory, detailed and final.

The introductory part is a sort of starting point to establish certain general issues.
The detailed part consists of questions that concern more specific details of the model.
Here a section for each Main Process Area (MPA) is present with question concerning
the MPA in question.
The final part consists of questions concerning what is lacking from the model, or if the
reader/interviewee feels that something should be added to the model.

Introductory part
1. What kind of position do you have in your company? Title?
2. How long have you been working in your current position? How long has RE
   been one of your responsibilities?
3. How long time did you spend studying the REPM Model?
4. What is the size of the company you are working for? (Number of employees,
   company turnover)
5. What is the size of the department/section you are working for? (Number of
   employees, department turnover)
6. What size is a typical project? (Time, man-hours, number of participants)
7. Are there people exclusively devoted to Requirements Engineering (RE) in a
   typical project? (How many people, how many man-hours)
8. In what way does RE enter in to a typical project?
9. How important would you say that the area of RE is pertaining to your projects?
   (Very Important, Average, Insignificant)
10. How do you feel that RE is involved in a software engineering project in general?
11. How important do you feel that RE is to a software engineering project in
    general? (Very Important, Average, Insignificant)
12. What would you say that Elicitation is within RE?
13. What would you say that Analysis and Negotiation is within RE?
14. What would you say that Management is within RE?

Detailed part
15. Is the meaning of what an Action easily understood from the model? How do you
    interpret the meaning of an Action?
16. Is the identification of Actions and Sub Process Areas (SPA) easy to understand?
    How do you interpret the division of SPA:s and Actions?
17. Is it easy to understand what the REPM Level aims to show when seeing the
    model? How do you interpret the intention of the REPM Levels?
18. Are the relations between different Actions easily understood?
19. What do you think determines which level an Action is on?
20. Do you feel that when looking at the different Actions on different levels they give a relevant picture of a RE specification and the RE Process?
21. Are there areas or Actions that you would grade differently concerning the REPM level?
   a. If answer YES – which ones and what grade would you give them?
22. Do you feel that one could grade the Actions in another fashion? In another scale or perhaps you have other suggestions?
23. Can you give a suggestion of how a project in total should be graded according to the REPM Model?

Elicitation
24. When looking at the Sub Process Areas under Elicitation do you feel anything is missing? (Is there a need for additional SPAs)
25. When looking at the Actions in the Main Process Area of Elicitation do you feel that everything is accounted for when it comes to Requirements Engineering?
26. Are the Actions placed correctly? (Under the correct MPA and SPA)
27. Do you feel that the Actions in the MPA of Elicitation are adequate?
28. Are there any parts you would like to take away completely?
29. How important do you feel that Elicitation is to a projects RE process? (Very Important, Average, Insignificant)
30. How common do you feel Requirement Reuse is within the RE process? (Very, Average, Unusual)
31. Do you feel that the SPA of Stakeholder Identification is covered adequately?
32. Do you feel that the SPA of Domain Knowledge is covered adequately?
33. Do you feel that the SPA of Stakeholder Consulting is covered adequately?
34. Do you feel that the SPA of Scenario Elicitation is covered adequately?
35. Is there something important missing according to you?
36. Have we missed to ask any other important questions?

Analysis and Negotiation
37. When looking at the Sub Process Areas under Analysis and Negotiation do you feel anything is missing? (Is there a need for additional SPAs)
38. When looking at the Actions in the Main Process Area of Analysis and Negotiation do you feel that everything is accounted for when it comes to Requirements Engineering?
39. Are the Actions placed correctly? (Under the correct MPA and SPA)
40. Do you feel that the Actions in the MPA of Analysis and Negotiation are adequate?
41. How important do you feel that Analyzing and Negotiation is to a projects RE process? (Very Important, Average, Insignificant)
42. How important would you say that Requirements Classification is to a projects RE process? (Very Important, Average, Insignificant)
43. What level of importance would you give the use of Interaction Matrices? (Very Important, Average, Insignificant)
44. Do you feel that the SPA of System Boundaries Definition is covered adequately?
45. Do you feel that the SPA of Requirements Prioritization is covered adequately?
46. Do you feel that the SPA of Requirements Risks is covered adequately?
47. Is there something important missing according to you?
48. Have we missed to ask any other important questions?

Management
49. When looking at the Sub Process Areas under Requirements Management do you feel anything is missing? (Is there a need for additional SPAs)
50. When looking at the Actions in the Main Process Area of Management do you feel that everything is accounted for when it comes to Requirements Engineering?
51. Are the Actions placed correctly? (Under the correct MPA and SPA)
52. Do you feel that the Actions in the MPA of Management are adequate?
53. How important is Requirements Management to a project's RE Process? (Very Important, Average, Insignificant)
54. Is Requirements Origin Specification easily understood from the model?
55. How would you describe the relation between Global System Requirements Identification and the mentioned domain knowledge areas under E.2?
56. What level of importance would you give to recording rejected requirements? (Very Important, Average, Insignificant)
57. What parts do you think a requirements document should consist of?
58. What importance do you give to recording requirements rationale? (Very Important, Average, Insignificant)
59. What level of importance would you give a standardized document structure has to a project and the RE process? (Very Important, Average, Insignificant)
60. Do you feel that the SPA of Requirements Document is covered adequately?
61. Do you feel that the SPA of CARE tool utilization is covered adequately?
62. Do you feel that the SPA of Traceability Policies is covered adequately?
63. Do you feel that the SPA of Requirements Change Policies is covered adequately?
64. What level of importance do you accredit the act of validating requirements? (Very Important, Average, Insignificant)
65. Is there something important missing according to you?
66. Have we missed to ask any other important questions?

Final part
67. Do you feel that there is something that is overlooked in the REPM Model?
68. What would you say are the main areas in the model?
69. Do you feel that there are other areas that should be present? (In addition to the MPAs covered)
70. Do you feel that the model is very easy to understand, easy to understand, neither easy to understand nor complicated, complicated or very complicated in terms of practical usability?
71. What do you think is the most complicated area to understand?
72. In what context would the model be used according to you?
73. What kind of problems can you see when trying to use/implement the model?
74. How well would the model suit in your current work process?
75. Do you have any additional comments?
Project Evaluation Checklist

Appendix III

Authors: Tony B. Gorschek
        Kaarina Tejle

Blekinge Institute of Technology
Department of Software Engineering and Computer Science
June 2002

© Tony Gorschek and Kaarina Tejle 2002
Permission to reproduce this document and prepare
derivative material for internal use is permitted provided the
copyright statement is included.
Requirements Engineering Process Model

*Project Evaluation Checklist version 1.4*

*(Action UID links each question to the relevant part in the REPM model)*

<table>
<thead>
<tr>
<th>E Requirements Elicitation</th>
<th>Action UID</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Do you reuse requirements from other systems developed in the same application area?</td>
<td>E.a1</td>
<td></td>
</tr>
<tr>
<td><strong>E.1 Stakeholder identification</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. When determining whom the stakeholders are for a system, do you ask the people ordering the system, whom they think are the stakeholders?</td>
<td>E.1.a1</td>
<td></td>
</tr>
<tr>
<td>3. Do you conduct your own research determining who the stakeholders are?</td>
<td>E.1.a2</td>
<td></td>
</tr>
<tr>
<td><strong>E.2 Stakeholder Consulting</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Do you consult the executive stakeholder (the people ordering/purchasing the system) in the elicitation process?</td>
<td>E.2.a1</td>
<td></td>
</tr>
<tr>
<td>5. Do you consult the larger general group of stakeholders in the elicitation process?</td>
<td>E.2.a2</td>
<td></td>
</tr>
<tr>
<td>6. Do you consult the in-house stakeholders, programmers, designers and managers of the production of the system, in the eliciting process?</td>
<td>E.2.a3</td>
<td></td>
</tr>
<tr>
<td><strong>E.3 Domain Knowledge</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Do you consider the influencing factors from the human domain area when eliciting requirements, e.g. political and organizational factors?</td>
<td>E.3.a1</td>
<td></td>
</tr>
<tr>
<td>8. When eliciting requirements do you study the application domain and the possible domain constraints from that area (not technical constraints)?</td>
<td>E.3.a2</td>
<td></td>
</tr>
<tr>
<td>9. Do you consider the systems operating environment when developing the system?</td>
<td>E.3.a3</td>
<td></td>
</tr>
<tr>
<td>10. Do you consider the larger picture, how the system is going to make a contribution to the organization, when eliciting requirements?</td>
<td>E.3.a4</td>
<td></td>
</tr>
<tr>
<td>11. Do you look at the business processes which the system being built should support?</td>
<td>E.3.a5</td>
<td></td>
</tr>
</tbody>
</table>
### E.4 Scenario Elicitation

<table>
<thead>
<tr>
<th></th>
<th>Question</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.</td>
<td>Do you use the technique of interaction sessions, known as scenarios when eliciting requirements?</td>
<td>E.4</td>
</tr>
<tr>
<td>13.</td>
<td>Do you consult the executive stakeholders for scenario elicitation?</td>
<td>E.4.a1</td>
</tr>
<tr>
<td>14.</td>
<td>Do you consult the general stakeholders for scenario elicitation?</td>
<td>E.4.a2</td>
</tr>
<tr>
<td>15.</td>
<td>Do you consult the in-house stakeholders for scenario elicitation?</td>
<td>E.4.a3</td>
</tr>
</tbody>
</table>
### A Requirements Analysis and Negotiation

<table>
<thead>
<tr>
<th></th>
<th>Action UID</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.</td>
<td>Do you have checklist with the steps to go through when analyzing the requirements?</td>
<td>A.a1</td>
</tr>
<tr>
<td>17.</td>
<td>Do you classify the requirements into classes or groups?</td>
<td>A.a2</td>
</tr>
<tr>
<td>18.</td>
<td>Do you apply the technique of Interaction Matrices?</td>
<td>A.a3</td>
</tr>
<tr>
<td>19.</td>
<td>If a requirement feels unclear/ambiguous do you make lists of what the requirement does not comprise?</td>
<td>A.a4</td>
</tr>
</tbody>
</table>

#### A.1 System Boundaries Definition

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20.</td>
<td>Have you in the analyzing stage of the project defined the system boundaries?</td>
</tr>
<tr>
<td>21.</td>
<td>Do you categorize the requirements (according to the boundary of the system) into different groups, e.g. System Requirements (inside the system scope), Requirements for the operational processes associated with the system and Requirements clearly outside the scope?</td>
</tr>
</tbody>
</table>

#### A.2 Requirements Prioritization

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>22.</td>
<td>Do you prioritize the requirements according to their importance to the stakeholders?</td>
</tr>
</tbody>
</table>

#### A.2.1 Requirements Re-prioritization

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>23.</td>
<td>Do you re-prioritize the requirements at any stage?</td>
</tr>
<tr>
<td>24.</td>
<td>Do you re-prioritize the requirements when new requirements are discovered?</td>
</tr>
<tr>
<td>25.</td>
<td>Do you re-prioritize the requirements when new releases come out?</td>
</tr>
<tr>
<td>26.</td>
<td>Do you re-prioritize the requirements with regularity?</td>
</tr>
<tr>
<td>27.</td>
<td>Do you re-prioritize the requirements when any kind of change occurs?</td>
</tr>
</tbody>
</table>

#### A.3 Requirements Risks

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>28.</td>
<td>Do you apply risk assessment in the step of analyzing the requirements?</td>
</tr>
<tr>
<td>29.</td>
<td>Do you analyze the risk for each individual requirement?</td>
</tr>
<tr>
<td>30.</td>
<td>Do you analyze the risk for sets of requirement?</td>
</tr>
<tr>
<td>31.</td>
<td>Do you analyze the risk for just a selected requirement?</td>
</tr>
<tr>
<td>M  Requirements Management</td>
<td>Action UID</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>32. Do you have a document that states the link between the natural language requirements to the models specifying the system?</td>
<td>M.a1</td>
</tr>
<tr>
<td>33. Have you identified the requirements that are only true for the system as a whole, the so-called Global system requirements?</td>
<td>M.a2</td>
</tr>
<tr>
<td>34. Have you documented which requirements that have been rejected and why?</td>
<td>M.a3</td>
</tr>
</tbody>
</table>

**M.1 Requirements document**

| 35. Do you uphold a document containing all the requirements? | M.1 | |
| 36. Does the document contain a specification of why the requirement is specified and what function the requirement has? | M.1.a1 | |

**M.1.1 Standardized Document Structure**

| 37. Do you apply a standard structure to all of your documents pertaining to Requirements Engineering? | M.1.1 | |
| 38. Does the requirements document contain a summary? | M.1.1.a1 | |
| 39. Does the requirements document contain a usage description, which describes the way to use the document referring to different groups of users? | M.1.1.a2 | |
| 40. Does the requirements document contain a business case showing the systems part in the business? | M.1.1.a3 | |
| 41. Does the requirements document contain a term definition part (dictionary for difficult words)? | M.1.1.a4 | |

**M.1.2 Describing Requirements**

| 42. Do you use a template to get a uniformed description of all the requirements? | M.1.2.a1 | |
| 43. Do you quantify (put a figure on) the non-functional requirements? | M.1.2.a2 | |
| 44. Do you use a clear and unambiguous natural language when you describe requirements? | M.1.2.a3 | |

**M.1.2.1 Descriptive Complements**
45. **Do you use diagrams (WBS, Gantt, Interaction diagrams and Architectural diagrams) as another way of structuring the information regarding requirements?**

### M.1.3 System Modeling

<table>
<thead>
<tr>
<th>Question</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>46. <strong>Do you use System Modeling to put requirements in their proper environment and to find new requirements?</strong> (Prototypes, System models, Environmental models and Architectural models)</td>
<td>M.1.3</td>
</tr>
<tr>
<td>47. <strong>Do you use prototyping to understand the requirements and to verify that they are specified?</strong></td>
<td>M.1.3.a1</td>
</tr>
<tr>
<td>48. <strong>Do you use system models e.g. data processing models, composition models, classification models, stimulus-response models and process models?</strong></td>
<td>M.1.3.a2</td>
</tr>
<tr>
<td>49. <strong>Do you apply the use of environment models, showing what lies outside the system?</strong></td>
<td>M.1.3.a3</td>
</tr>
<tr>
<td>50. <strong>Do you apply the use of architectural models showing the entire system, sub-systems and the links between them?</strong></td>
<td>M.1.3.a4</td>
</tr>
</tbody>
</table>

### M.1.4 Requirements Validation

<table>
<thead>
<tr>
<th>Question</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>51. <strong>Do you apply the technique of Requirements inspections?</strong></td>
<td>M.1.4.a1</td>
</tr>
<tr>
<td>52. <strong>Do you conduct requirements reviews during the Requirements Engineering process?</strong></td>
<td>M.1.4.a2</td>
</tr>
<tr>
<td>53. <strong>Do you apply the technique of System Model Paraphrasing – taking system models and converting them into natural language requirements?</strong></td>
<td>M.1.4.a3</td>
</tr>
<tr>
<td>54. <strong>Do you develop a user manual draft at an early stage in the process?</strong></td>
<td>M.1.4.a4</td>
</tr>
<tr>
<td>55. <strong>Do you apply the technique of having a test case for each requirement?</strong></td>
<td>M.1.4.a5</td>
</tr>
</tbody>
</table>

### M.2 CARE Tool Utilization

<table>
<thead>
<tr>
<th>Question</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>56. <strong>Do you use CARE tools for information interchange?</strong></td>
<td>M.2.a1</td>
</tr>
<tr>
<td>57. <strong>Do you use CARE tools for information handling?</strong></td>
<td>M.2.a2</td>
</tr>
</tbody>
</table>

### M.3 Traceability Policies

<table>
<thead>
<tr>
<th>Question</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>58. <strong>Do you have a unique identifier for each requirement?</strong></td>
<td>M.3.a1</td>
</tr>
<tr>
<td>59. <strong>Can you trace (through your documentation) the origin of a certain requirement in people and/or other documents?</strong></td>
<td>M.3.a2</td>
</tr>
</tbody>
</table>
60. Do you document the link between the requirements and the design and implemented components?  
   M.3.a3

61. Do you document the link between design and implemented components back to their requirement(s)?  
   M.3.a4

62. Do you link preceding documents (e.g. pre-study documentation) to the relevant requirements?  
   M.3.a5

63. Do you have version traceability of requirements?  
   M.3.a6

<table>
<thead>
<tr>
<th><strong>M.4 Requirements Change Policies</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>64. Do you identify the volatile requirements, the ones likely to change during the process?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>M.5 Documentation Deliverables</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>65. Do you define what documentation shall be delivered with the system?</td>
</tr>
<tr>
<td>66. Do you define what user documentation should be delivered?</td>
</tr>
<tr>
<td>67. Do you define what system documentation should be delivered?</td>
</tr>
<tr>
<td>68. Do you define what management documentation should be delivered?</td>
</tr>
</tbody>
</table>