
Process Improvement Proposals in System Requirements Management - an Industrial Case Study

Bachelor of Science Thesis

Urban Martinsson

Åsa Karlsson

C Technologies AB

Department of Communication Systems

Lund Institute of Technology, Lund University

Abstract

Requirements engineering is the process of discovering, documenting and maintaining requirements. To manage this in a structured and repeatable manner, it is preferable to follow a process, with defined phases, activities and actors, instead of just relying on the employees' competence and personal ability.

C Technologies AB is a young company, developing technical advanced products containing both software and hardware. They are of the opinion that their current requirement engineering process could be improved to better serve their needs. Improving the process is the main goal for this thesis. In order to achieve this a survey of their current process has been conducted. Different process models have been studied and a process improvement proposal has been made. A database prototype that supports the using of the proposed process has also been developed.

The process proposal defines actors, stakeholders, their participation and responsibilities. It also defines a number of phases and activities, which may be used in future requirement engineering at C Technologies. If the process is deployed C Technologies wants to be able to evaluate the process proposal. By using the GQM method a process evaluation plan has been produced, which defines the metrics that should be collected.

Preface

Our work at C Technologies the last four months has resulted in our Bachelor Thesis. The interest we had gained for Requirements Engineering during our education has been further developed and used. There is a lot of information to be found regarding Requirements Engineering and it was a very interesting work to find the most suitable parts for C Technologies.

As we started we had pretty clear goals with the thesis and we think we have succeeded in reaching them. We hope that our work will gain C Technologies in the future and that the persons reading the thesis will find it interesting and educating.

Acknowledgements

First of all we would like to thank Per Beremark, our supervisor at C Technologies, for his support and participation.

We would further like to thank Johan Natt och Dag and Björn Regnell, our academic supervisors from the Department of Communication systems, Lund Institute of Technology for their interest and participation in our work.

Also, thanks to Ofelia Madsen, Usability Architect at C Technologies, whose effort and help with all sorts of things has been very important and appreciated.

Ola Håkansson and personnel at C Technologies, who took the time to answer our questionnaire and thereby provided information for our survey.

Everyone else at C Technologies, who has somehow given input to our thesis.

Urban Martinsson

Åsa Karlsson

Lund, Helsingborg
June 2001

Table of contents

Abstract.....	3
Preface.....	5
Acknowledgements	5
1 Introduction.....	9
1.1 Background.....	9
1.2 Objectives	9
1.3 Limitations.....	9
1.4 Thesis Overview	10
2 Requirements Engineering.....	11
2.1 Quality	11
2.2 Process improvement.....	12
2.2.1 CMM – The Capability Maturity Model	13
2.2.2 ISO 9000.....	15
2.2.3 Software Measurements and Metrics.....	15
2.3 Requirements	16
2.4 Requirement engineering activities	17
2.4.1 Feasibility study	17
2.4.2 Requirements Elicitation	18
2.4.3 Requirements Documentation.....	20
2.4.4 Requirements Validation	23
2.4.5 Requirements Prioritization	24
2.4.6 Requirements change management	25
2.5 Requirements Engineering Processes.....	26
2.5.1 A general requirements engineering process model	27
2.5.2 The spiral model	27
2.5.3 State-oriented model	28
2.6 Computer Aided Software Engineering tools.....	29
3 Current situation analysis	31
3.1 Survey.....	31
3.1.1 Purpose.....	31
3.1.2 Performing the survey.....	31
3.1.3 Analysis of survey	33
3.2 Result.....	36
3.2.1 Conclusions.....	36
3.2.2 Current situation at C Technologies	37
3.2.3 C Technologies future needs.....	39
4 Process Improvement Proposal	41
4.1 Our Process proposal.....	42
4.1.1 Actors and their responsibilities	43
4.1.2 Process phases.....	45
4.1.3 Requirement states	49
4.1.4 Process activities.....	53
4.1.5 Requirement attributes	56

5	Process realization.....	59
5.1	Database prototyping.....	59
5.1.1	The actual database prototype.....	61
5.2	Requirement Scenarios.....	65
6	Process Proposal Evaluation Plan	73
6.1	Background.....	73
6.2	The Goal-Question-Metric method	73
6.2.1	Goals	74
6.2.2	Questions	74
6.2.3	Metrics	76
6.3	Metrics recommendation	77
7	Summary and Further Work.....	81
7.1	Summary.....	81
7.1.1	Process Improvement Proposal.....	81
7.1.2	Process Proposal Evaluation Plan.....	82
7.2	Further work	83
8	Glossary.....	85
9	References	91
Appendix A.....	95	
	CMM – Requirements Management	95
Appendix B.....	97	
	Questionnaire.....	97
Appendix C.....	115	
	Requirement specification	115
Appendix D.....	119	
	Process proposal reference version.....	119
Appendix E	127	
	Review checklist.....	127
Appendix F	129	
	Database prototype	129
Appendix G.....	131	
	Prototype implementation code	131

1 Introduction

The introduction presents the background, objectives, and limitations of this Bachelor thesis. It also gives the reader an overview of how the thesis is disposed.

1.1 Background

Being able to perform successful development projects includes work in several different areas. Technical advanced products containing software has become a huge industry and there are many competitors on the market. One of the most important parts of the development process is requirements engineering. The process of developing new products always starts with some kind of wishes or needs. The wishes and needs can be of help finding requirements that describe what the product shall be, look like and what it shall be able to perform. Discovering, documenting and maintaining requirements is often defined as requirements engineering [Sommerville, 2001].

C Technologies AB is a very young company, developing technical advanced products containing both software and hardware. C Technologies is the main company in a group of companies consisting of WeSpot, Anoto and C Technologies. The entire organization has around 250 employees and most of their products are directed to customers like students and business-people.

C Technologies are of the opinion that their current requirement engineering process could be improved to better serve their needs. In order for this they want to perform a study of their current requirement engineering process in order to find weaknesses, strengths and improvement areas. They also want an improvement proposal, based on the result of the survey, containing the most relevant activities and methods suited for their requirement process.

1.2 Objectives

The Bachelor thesis aims at achieving the following objectives:

- To give the reader an introduction to Requirements Engineering
- To make an investigation and analysis of C Technologies current work and future needs regarding their requirement process
- To provide C Technologies with a Process Improvement Proposal that considers their special needs
- To provide C Technologies with an Evaluation Plan for the proposed process

1.3 Limitations

Due to the time limits of the Bachelor Thesis some limitations has been made. The following list contains issues that are not considered in the survey or in the Process proposal.

- Requirements change management
- How to present the requirements in the specification, for example by use cases or context diagrams
- Evaluation of the Process Proposal

Even though these limitations are made, some information about them is presented in chapter 2. This is because we want to provide the reader with an overall picture of Requirements Engineering, its activities, methods and procedures, for easier understanding of the report.

1.4 Thesis Overview

Since the thesis is a part of a Bachelor of Science in Software Engineering, the intention is that students with similar education shall be the main audience. We also hope that personnel at C Technologies and other persons interested in Requirements Engineering, shall be able to understand and find it useful.

The thesis is divided into following chapters:

1. **Introduction** describes background, objectives and limitations with the thesis
2. **Requirements Engineering** gives an introduction to several different theories and aspects of RE.
3. **Current Situation Analysis** presents how the investigation was performed and the results from the analysis of the investigation.
4. **Process Improvement Proposal** describes the work with the proposal and presents the conclusions and results and how to apply the process at C Technologies.
5. **Process Proposal Evaluation Plan** describes metrics and instructions for how to measure and evaluate the proposed process.
6. **Summary and Further Work** presents a summary of the thesis and possible new work areas that the thesis has led to.

2 Requirements Engineering

The purpose of this chapter is to present requirements engineering, what it is, how it can be used and why it should be considered important. Since our background is in software engineering this is the foundation for the report. It is however important to know that requirements engineering can be applied to all different kinds of developing companies.

In general creating new products with high quality, demands both time and effort. It starts with an idea or a need of some kind. The idea shall be evaluated and performing the evaluation involves requirements engineering. It is important to specify the ideas or the needs in a way that makes the system or product to be developed understandable. Requirements engineering helps us to do this in an organized and structured way. If performed properly it may reduce development time and effort, and at the same time increase the quality. Probable stakeholders involved in requirements engineering are developers, customers, end users, personnel from the marketing department and other parties interested in the product or system to be built [Sommerville, 2001]. Ideas, opinions and wishes should be collected from all stakeholders. Requirements engineering is however not all about collecting ideas. It also includes documenting and validating them. Since it usually is impossible to realize all requirements, it is common to prioritize them. The common activities in requirements engineering will be explained and discussed further in this chapter.

One of the major reasons for practicing requirements engineering is to gain higher quality. Therefore the concept of quality will be discussed in the following section.

2.1 Quality

Quality can be defined in several ways since there are many different aspects that may be considered. One definition of quality is that the system or product meets its specification [Sommerville, 2001]. Another definition of quality is “*Quality is what the customer says it is*” [Feigenbaum, 1994]. Working with quality issues in a company is both important and demanding, there are always procedures and activities to improve and there is no such thing as a perfect organization. A main goal for every company should be to deploy quality work throughout the entire organization. On today’s market, high quality products are one of the biggest advantages when it comes to competition.

There are several actions that may be taken in order to achieve higher quality in an organization. Working with process improvements, standards and guidelines are the most common ones. Writing a quality plan, and deploying it, is one way to introduce standards and guidelines. A quality plan should at least include directions for the activities that need to be performed. The quality work needs to be managed, and according to [Sommerville, 2001] there are three principal activities that can be of help structuring the work:

1. **Quality Assurance** is about establishing frameworks of procedures and standards throughout the organization. The procedures and standards will help structure the work, which will lead to improved software quality.
2. **Quality Planning** is about finding suitable procedures and standards for the specific organization and applying them to the work. There can be both a general quality plan and specific quality plans for certain projects.
3. **Quality Control** is about following up the work accomplished, and make sure that the development team follows the procedures and standards as planned.

One of the steps towards higher quality is well-performed requirement engineering. For a company there are different ways to find suitable procedures and guidelines about how to achieve this. A great amount of literature about tools, processes and standards exists, in which a quality manager can find advice about what actions can be taken. It is however important to be aware of, that even if standards and procedures have been introduced and are used in the company, there will always be improvement areas.

2.2 Process improvement

Today more and more companies work with development processes. There is a general opinion that using development processes facilitates the work, makes it easier to meet deadlines and produce high quality products [Sommerville, 2001]. However, the processes can always be better, more effective and more tuned for every specific company. This makes working with continuous improvement important. Working with process improvement is a long and iterative process. Improvements often have to be made on several different activities in the development process. These improvements shall not be made simultaneously. Instead the steps towards process improvement have to be taken one at a time. The positive effects of process improvements are that they can help companies reduce development time and costs, meet deadlines, finding errors early etc, which will lead to increased competitiveness. As figure 2.1 shows there are a number of key stages in a process improvement process.

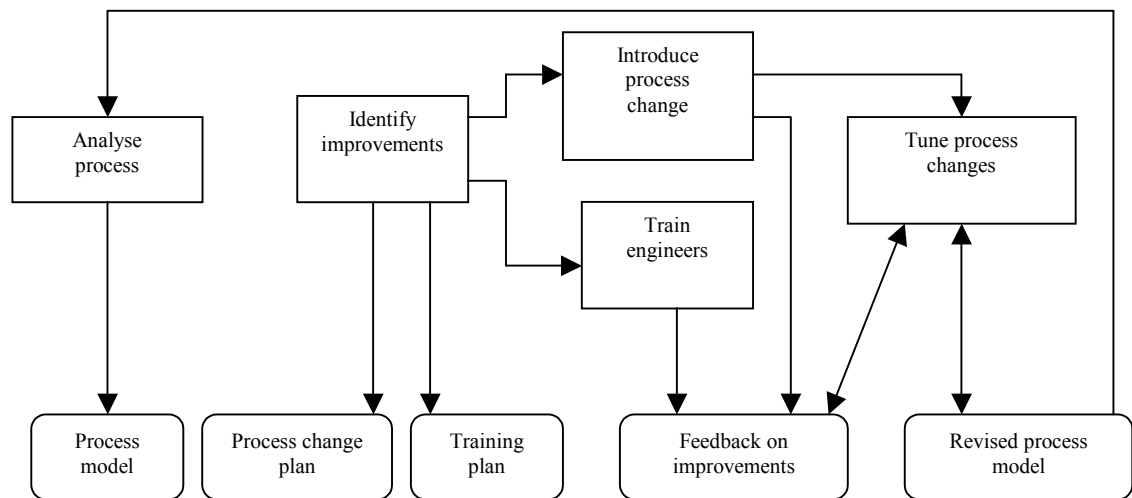


Figure 2.1 Process Improvement model [Sommerville, 2001].

1. **Process analysis**, current processes must be analyzed and documented.
2. **Improvement identification**, the process analysis is used to find weaknesses and bottlenecks in the currently used processes. The stage includes proposing new procedures and methods to address the problems.
3. **Process change introduction** introduces the new proposed procedures and methods and integrates them with the already existing ones.
4. **Process change training**, it is important that all personnel work towards the new goals and to train them in how the new process works is essential.
5. **Change tuning**, the chance of immediate improvement is small. The new processes have to be tuned, modified and adjusted to reach the level of wanted performance.

Higher quality may be achieved by introducing process improvement into an organization. The next two chapters present two common improvement models, the CMM – the Capability Maturity Model and the ISO 9000 standard.

2.2.1 CMM – The Capability Maturity Model

The Capability Maturity Model was developed by SEI – the Software Engineering Institute at Carnegie Mellon University in the mid eighties. At this time it was common with software catastrophes. Many projects were late and over budget [Paulk et al, 1993]. The US Department of Defense, among others, reacted strongly to this and demanded higher quality from their suppliers. This led to the development of CMM.

CMM is a framework that describes key elements of an effective software process. The main idea is to focus on a limited set of activities, and try to improve them. It guides developers on how to effectively gain control over their processes. It also helps them to select a process improvement strategy by determining the maturity of the organization.

The purpose of the Capability Maturity Model is to help organizations reach a higher level of maturity. It consists of five maturity levels where the first is the *Initial* level and the fifth is the *Optimizing* level as shown in figure 2.2. To reach a higher level an organization takes many small steps. The CMM provides a framework for the steps to enable process improvements.

Each of the five maturity levels consists of their own Key Process Areas. These indicate on which areas an organization should focus their improvement efforts. The Key Process Areas identify the issues that have to be handled to achieve a maturity level.

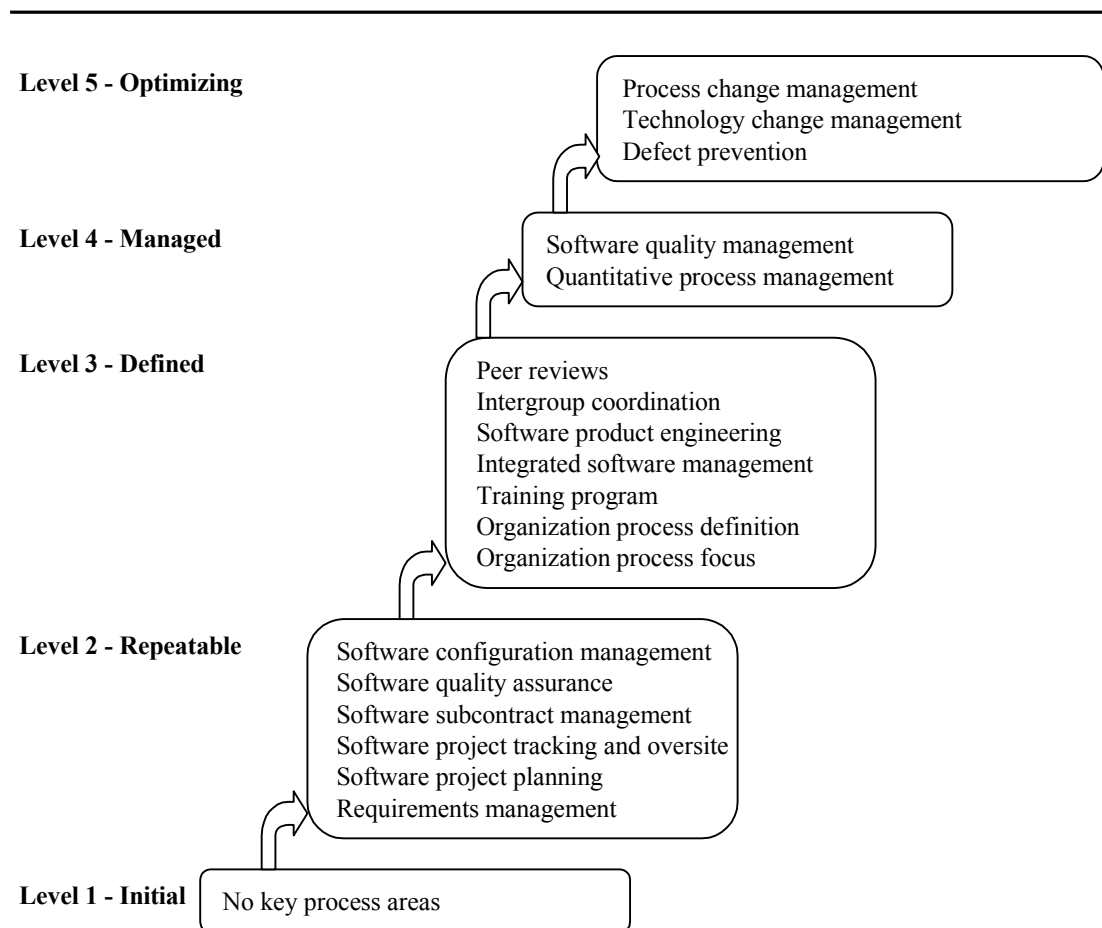


Figure 2.2 Key process areas of CMM

CMM addresses many activities within the software process. Examples of focusing points are Configuration Management, Quality assurance, Project planning, Contract

Management, Training program, Defect prevention and Requirements Management. The CMM activities that specifically address Requirements Management are presented in Appendix A.

2.2.2 ISO 9000

The original intention of the ISO 9000 standards was to replace the many quality standards worldwide with a single set of common quality standards. The theory was that unified standards would reduce barriers to trade, by replacing individual country standards with a single set of global standards. ISO 9000 is the result of companies needing to hold on to or secure new business from other companies that are requiring ISO 9000 registration as a condition for doing business.

ISO 9001 is a model for quality assurance in design/development, production, installation and servicing. It specifies quality system requirements to be used when a company needs to demonstrate its ability. ISO 9001 states 20 points from Management responsibility and Contract review to personnel training [Steeple, 1994].

The emphasis of ISO 9000 is the documentation and implementation of quality procedures and quality records. This implies that the main purpose of the standard is to ensure that basic quality systems are in place. The quality discipline in ISO 9000 requires that a company define, document and implement quality procedures.

When implementing models like CMM or ISO 9000 it is important to keep track of the changes. Measuring and analyzing through metrics collection may help this.

2.2.3 Software Measurements and Metrics

Measurements are made mainly because of three things [Fenton, Pfleeger, 1997]:

- We want to understand what happens during development and maintenance
- We want to control what is happening in our projects
- We want to improve our current processes

To be able to determine whether changes on the processes have led to improvement or not it is useful and sometimes even necessary to use software metrics. An example of a software metric is to count how many errors that are found in the code [Humphrey, 1990]. When the errors are counted they can be analyzed to see where they derive from. Perhaps it turns out that if a better design had been made not so many errors would have occurred. There must be a well-defined goal and motivation for every metric. It is important that collecting and analyzing the metrics is not too demanding. The focus must still be on developing new products. The right amount of metrics must

be decided for each company, only the most valuable ones should be used and if possible they should be collected automatically.

In a project, management and developers often are considered as stakeholders. Management need to measure how much a software process costs, how productive the staff is, how good the code being developed is and other things that help them estimate development costs, measure the quality of the products, predict development time and so on. The developers' approach differs from the managements'. The developers can use metrics to analyze if requirements are testable, predict remaining faults and determining if the product or process goals have been met etc [Fenton, Pfleeger, 1997]. The developers can also measure their individual improvement, for example how their programming skills improve, how they may get faster and make less faults.

Process improvement aims at improving different activities and methods. Before we introduce requirements engineering methods and activities requirements will be explained and defined.

2.3 Requirements

There are several definitions of what a requirement is. Most likely it is impossible to exactly define requirements and what they shall include since this varies depending on the system or product being developed but two examples are:

“A requirement is something that the product must do or a quality the product must have” [Robertson, Robertson, 1999].

“Requirements are descriptions of how the system should behave, application domain information, constraints on the system's operation, or specifications of a system property or attribute” [Kotonya, Sommerville, 1997].

Requirements are often categorized and used on different levels in the requirement process. This is done since often a large number of requirements are elicited and grouping them makes them easier to handle. The two most common categories are functional requirements and non-functional requirements. These are defined below together with a definition of high-level requirements.

Functional requirements

“The functional requirements specify what the product must do. They relate to the actions that the product must carry out in order to satisfy the fundamental reasons for its existence” [Robertson, Robertson, 1999]. An example of a functional requirement is:

With the hotel booking system it shall be able to register a new booking.

Non-functional requirements

Non-functional requirements are requirements, which are not specifically concerned with the functionality of a system. The non-functional requirements define the overall qualities and attributes of the resulting system or product. Examples of non-functional requirements are security, safety, reliability and performance requirements [Kotonya, Sommerville, 1997]. Usability requirements is another type of non-functional requirements that often is considered to be a separate category of requirements. Usability requirements define how the product shall be presented to the user. The importance of developing products that are easy to use and understand makes it necessary to express requirements of this kind [Kotonya, Sommerville, 1997].

High-level requirements

High-level requirements derive from slogans and ideas about the system with a low level of detail. They contain what the system shall be able to perform, what functions it shall include, and in what environment it shall operate etc. One way to decide if a requirement really is a high level requirement is to look at the possibility to estimate the time effort needed to implement it. If the estimated time exceeds several months it is probable that it is a high-level requirement and that it has to be broken down into smaller parts to make the estimation more accurate.

Next section describes common requirement engineering activities.

2.4 Requirement engineering activities

Working with requirements engineering is a process, which contains several activities. The process is often presented in a model. There are several process models but some common activities are almost always included. Before we present examples of requirements engineering process models we will describe some of the common activities.

2.4.1 Feasibility study

Whenever an idea about a new product or system comes up it is preferable to make a feasibility study [Sommerville, 2001]. The purpose of the study is to investigate the idea. The feasibility study can result in a description, of the product or system, presented in a report. The report shall recommend and motivate whether or not it will be profitable to start up a project.

Answering the following questions might be of help when deciding on new products to be developed.

1. Does the system or product contribute to the overall objectives of the organization?
2. Can the system or product be implemented using current technology and within given cost and schedule constraints?

3. Can the system or product be integrated with other systems/products that are already in place/existing?

Managers, developers, technical experts, end-users and other stakeholders shall all be part of the feasibility study and answer the questions.

2.4.2 Requirements Elicitation

Elicitation is about finding and formulating requirements for a system or product [Lauesen, 2000]. The purpose of the requirements is to define what functions the system or product to be developed shall have. The requirements should also define what the system or product shall look like and what performance capability it shall have. Different types of requirements will be discussed later in the report. Stakeholders in the elicitation might be customers, development teams, marketing departments, management and others. All these people probably have opinions that differ a lot, but in the eliciting phase of requirements engineering it is important to gather and consider all viewpoints and possibilities.

There are some issues that complicate the elicitation. Often, the stakeholders do not recognize their own needs, which leads to requirements being missed or forgotten [Lauesen, 2000]. Another thing to be aware of is that the demands on the system or product change over time. It is difficult to know when to stop accepting that the stakeholders keep changing their minds about the requirements, but at some point the elicitation phase has to end.

To be able to find requirements it is important to understand the domain in which the system or product shall operate [Lauesen, 2000]. Depending on what kind of system or product that will be developed the domain will vary. Therefore it is good to perform a domain investigation in order to increase the knowledge about the domain. An example is; when developing a new version of a C Pen (C Pen is C Technologies main product) it is important to examine how the C Pen's that are on the market today works, and what changes would be preferable. Maybe some functions are useful as a base for new features and out of this knowledge it will be easier to elicit new requirements. Another example is the development of a whole new booking system for a hotel. Important to know is how bookings are performed today and what the new system could provide to facilitate the work. The goal of the domain investigation is to make a list of the present problems in the domain and thereby find critical issues and goals for the system or product, collecting ideas and realistic possibilities about the new system and make requirements out of them [Lauesen, 2000].

How the elicitation is performed differs between companies and projects. Some companies or projects might have an ongoing gathering of ideas and high-level requirements, while others elicit through specific methods at settled occasions. For the

latter case there are various methods accessible. Below, examples of elicitation methods are presented together with a short explanation.

Brainstorming

The idea of brainstorming is to let the participants, stakeholders of all kind, express all kinds of ideas about the system or product to be developed [Lauesen, 2000]. All ideas are documented and no criticism is allowed. Since all ideas are welcome there will probably be unrealistic wishes, so before ending the brainstorming session it is common to prioritize the elicited ideas and requirements.

Focus groups

A focus group is a more structured form of brainstorming. The participants start with expressing problems in their current situation with the system or product and after that they try to come up with the ideal way to handle the problems [Lauesen, 2000]. Explaining why their new ideas are good will help them formulate goals for the new system or product. The stakeholders are divided into groups and at the end of the meeting each group prioritizes some of their requirements. In later prioritization it is important to keep some requirements from all focus groups so that all stakeholders are satisfied.

Prototyping

“A prototype is a simplified version or part of the final system” [Lauesen, 2000]. Often users find it hard to express requirements without being able to visualize the system or product. Letting stakeholders experiment with a prototype can help them find new requirements. By testing parts of the product they can easily find out what functions are missing and determine whether the user interface needs improvement or not [Kotonya, Sommerville, 1997].

Scenarios

It might be useful to develop scenarios when eliciting ideas and requirements. End users and stakeholders often find it easier to relate to real-life examples rather than abstract descriptions of the functions provided [Kotonya and Sommerville, 1997]. By letting end-users simulate their interaction with systems or products using scenarios, more ideas and requirements may be thought of.

Goal-means analysis

Goal-means analysis is more of a checking technique than an elicitation method. The aim of the technique is to ensure that no goals are forgotten and that the final system meets them. Another reason for using goal-means analysis is that it supports prioritization by emphasizing the purpose of the requirements.

The goal-means analysis looks at the relation between goals, issues and requirements [Lauesen, 2000]. In general the analysis answers the following questions:

1. For each high-level system goal, are there issues and requirements that ensure that the goal can be met?
2. For each requirement, is it explained what its purpose is?
3. Is the requirement on the right level or should the issue or goal be the requirement?

2.4.3 Requirements Documentation

The documenting phase starts when eliciting the requirements. As soon as new requirements are found they must somehow be saved. It is important that requirements do not disappear or are forgotten. Storing all requirements for example in a database enables and facilitates the producing of traceable requirements. The point of being able to trace a requirement is to be able to see what has happened to it, if it has been rejected and why, if it already has been implemented or moved to another release etc. Tracing a requirement also includes keeping track of whoever came up with the idea at first.

To make the requirements understandable to their readers they must be presented in a suitable way. Søren Lauesen describes several ways to present requirements in his book “Software Requirements – Styles and Techniques”. Below some of them are explained. In all examples the product to be developed is a hotel booking system.

The *feature style* is the most common way of presenting requirements. With this method functional and non-functional product properties are explained in plain text. An example of a feature style requirement follows:

The hotel booking system shall be able to store 500 bookings.

Often these requirements are formulated with a “shall” phrase. This clearly states that it is a requirement. The problem with feature style requirements is that they are difficult to formulate if we want them to be completely unambiguous.

A *context diagram*, see figure 2.3 shows requirements on the product to be developed. It also shows how it communicates with user groups and external systems. This gives the reader a good overview of the system and it is easy to see if requirements are missing. It also supports the decision making about what shall be included and what shall be left out. The context diagram may be used as a checklist for what to develop and it makes it easy to verify that all requirements have been implemented.

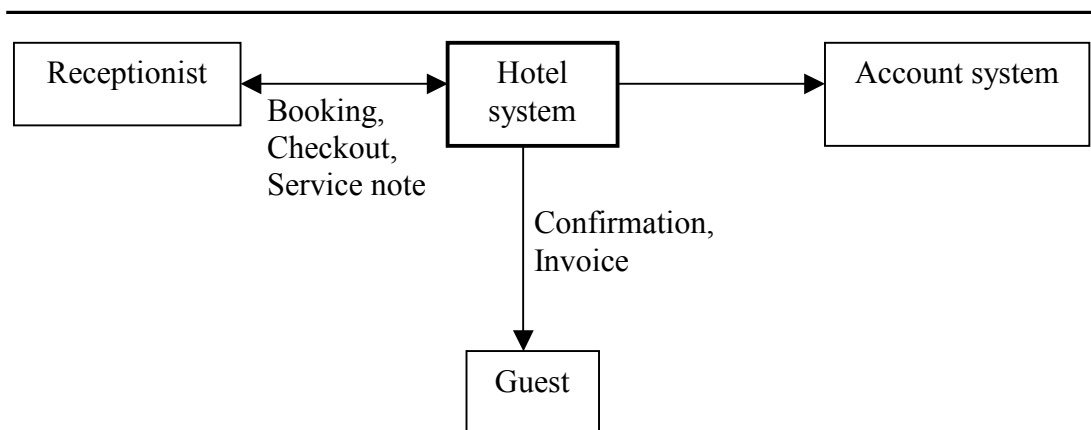


Figure 2.3 Hotel system context diagram [Lauesen, 2000]

The *data model*, figure 2.4, shows the relation between data in the system. The model is very effective when it comes to showing the data that has to be stored in a system or product, and is therefore suitable when modeling relational databases. When producing a diagram of this kind it is important to complement it with natural language that further explains the entities and attributes.

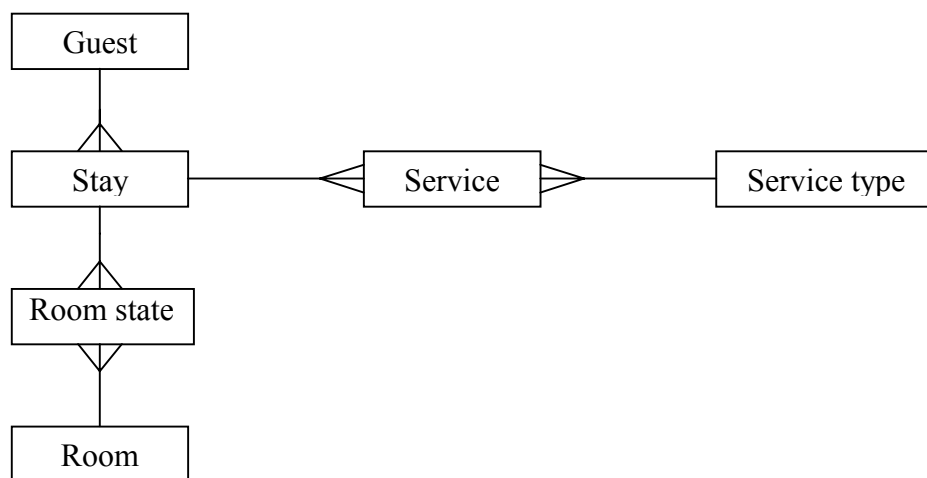


Figure 2.4 Hotel system data model [Lauesen, 2000]

When trying to explain and decide which data a system uses and produces, it is preferable to use a *data-flow diagram*, figure 2.5. It shows activities triggered by events produced by the system. These diagrams can be used at different abstraction levels, that is, the amount of information and details may vary depending on the diagram's purpose. For example is it possible to produce a diagram at **product level** that describes the general functions to be provided by the product.

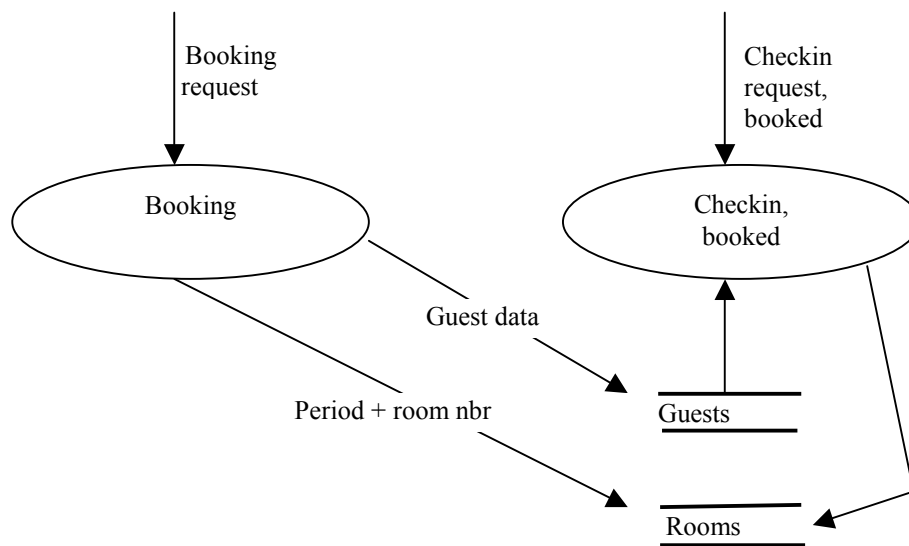


Figure 2.5 Hotel system dataflow diagram [Lauesen, 2000]

Another way of presenting requirements is by *use cases*, figure 2.6. A use case describes an activity carried out by a user of the system or product. Use cases can be designed in different ways. One example is to present them as the UML (Universal modeling language) notation does. UML is a standard in object-oriented development, which includes using use cases. Another way of expressing use cases is with *task notation*, which means describing the ongoing activities in a domain in a partly structured text form. This method can be extended to contain problems and solutions for the activities described.

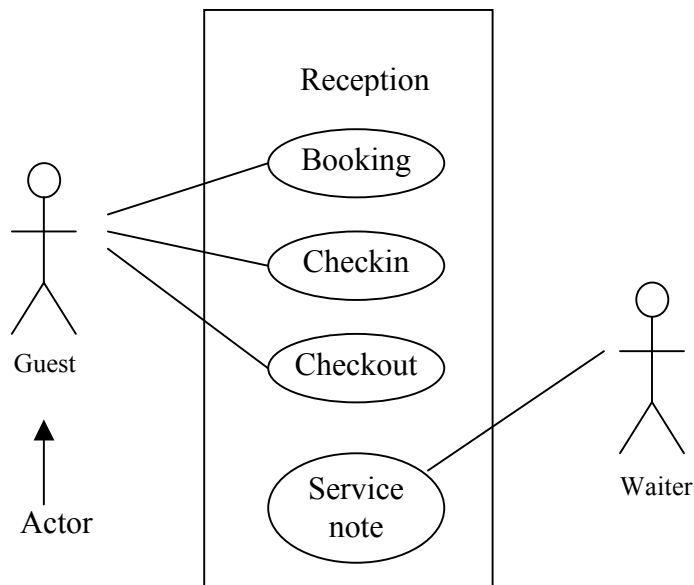


Figure 2.6 Hotel system use case [Lauesen, 2000]

Different ways of expressing requirements are suitable for different requirements. Some methods are suitable for functional requirements and other for usability requirements. Context diagrams and data-flow diagrams are usually easy for stakeholders and customers to understand, even if they have no earlier experience reading diagrams like these. The customer often prefers feature style since it uses natural language, which makes it possible for them to easily produce requirements.

2.4.4 Requirements Validation

Requirements that are documented can be validated. The purpose of validating requirements is to make sure they fulfill certain quality attributes. Possible attributes are presented below. The validation can be performed in various ways but the most common is to make a review of the specification.

Traceability

Include links to related requirements and to the reasons why these requirements have been included? Is there a clear link between software requirements and more general systems engineering requirements? [Kotonya, Sommerville, 1997]

Correctness

A requirements specification is correct if and only if every requirement stated therein represents something required of the system to be built [Davis, 1993].

Ambiguity

Are the requirements expressed using terms, which are clearly defined? Could readers from different backgrounds make different interpretations of the requirements? [Kotonya, Sommerville, 1997]

Verifiability

A requirements specification is verifiable if every requirement stated therein is verifiable. A requirement is verifiable if a person or machine can check that the built product meets its specification [Davis, 1993].

Understandability

Can readers of the document understand what the requirements mean? This is probably the most important attribute of a requirement document – if it cannot be understood the requirements cannot be validated [Kotonya, Sommerville, 1997].

Modifiable

A requirements specification is modifiable if changes to the requirements can be made easily, completely and consistently [Davis, 1993].

Redundancy

Is information unnecessarily repeated in the requirements document? Sometimes of course, repeating information adds to understandability. There must be a balance

struck between removing all redundancy and making the document harder to understand [Kotonya, Sommerville, 1997].

Completeness

Does the checker know of any missing requirements or is there any information missing from individual requirement descriptions? [Kotonya, Sommerville, 1997]

Consistency

Do the descriptions of different requirements include contradictions? Are there contradictions between individual requirements and overall system requirements? [Kotonya, Sommerville, 1997]

Testability

Determining whether or not the requirements can be tested is a way to determine if they are accurate. All requirements must be testable. An advantage with this is that it gets the testers involved in the project at an early stage.

Organization

Is the document structured in a sensible way? Are the descriptions of requirements organized so that related requirements are grouped? Would an alternative structure be easier to understand? [Kotonya, Sommerville, 1997]

Conformance to standards

Does the requirements document and individual requirements conform to defined standards? If there is a department of standards, is it justified? [Kotonya, Sommerville, 1997]

2.4.5 Requirements Prioritization

When a validated requirement specification is presented it often contains a large number of requirements. Prioritizing among them may then be necessary since it in many cases is impossible to implement all the requirements and still deliver a high-quality product [Wiegers, 1999]. There might be too many requirements, or some requirements might be too expensive or take too long time to implement, to be practicable for the current system release. It is important to deliver the most essential function as early as possible.

Prioritizing requirements should include several stakeholders. Depending on what kind of product or system it is to be developed and what purpose it has the stakeholders can vary. It is common that some stakeholders have more impact on the outcome than others. Negotiations are often held since it is important that all stakeholders get to express their needs and motivations to why one requirement is more important than another.

Different methods may be used when prioritizing the requirements. Cost-Value based prioritization through pair-wise comparison [Karlsson, 1996] and absolute numerical

analyze where different prioritization scales are used [Wiegers, 1999] are two examples which will be described.

Absolute numerical analysis

Using a numerical analysis method means that the stakeholders get to give each requirement a priority number by considering the importance of the requirement.

An example of a three-level scale is presented below:

- 3 Must be implemented perfectly
- 2 Needs to work, but not spectacularly well
- 1 Can contain bugs

Cost-Value based prioritization through pair-wise comparison

When using the pair-wise comparison technique two requirements are compared from out a predefined question or statement. The question to bear in mind when comparing two requirements could be:

With emphasis on security, which of these two requirements is most important?

The technique does not only include deciding which requirement is the most important, but also, on a scale show, how much more important one is compared to another. With this technique it is rather easy to find a number of requirements that are suitable for implementation.

The final document that contains the requirements to be implemented for a product or release is often called requirements specification. A requirements specification should at least describe services and functions, overall properties and interaction with other systems.

2.4.6 Requirements change management

In requirements engineering it is also important to be aware that there will be requests for changes on the requirements. Both customers and project members can come up with change requests and these must somehow be considered. Changes can be made to a single requirement, it can also mean that requirements get added or removed to the requirements specification. The important issues to consider are the following:

- How will the change affect the product?
- How will the change affect the surroundings? For example, what happens if a change is made on a requirement on which other requirements rely?

To be certain about the consequences of a change it is recommended to have a controlled process for changing requirements. An organization that finds it important to control their processes should ensure that [Wiegers, 1999]:

- Proposed changes are carefully evaluated
- The appropriate individuals make decisions about changes
- Changes are communicated to all affected participants
- The project incorporates requirements changes in a disciplined fashion

One way to implement a change management process is to use a simple state machine through which a requirement passes when it goes through change [Wiegers, 1999]. It is also recommended to have a change control board that is responsible for making decisions about what changes shall be made. The change control board shall include persons that are well familiar with the project and have good knowledge about the product to be developed.

When defining a change control process it is important the project management has communicated a policy that states how requirement change requests are supposed to be handled. The following elements of a change control policy have been found to be helpful [Wiegers, 1999]:

- All requirements changes must follow the process. If a change request is not submitted in accordance with this process, it won't be considered
- Simply requesting a change does not guarantee that it will be made. The project's change control board will decide which changes to implement
- The contents of the change database must be visible to all project stakeholders
- The original text must not be modified or deleted from the database
- Every incorporated requirement change must be traceable to an approved change request

There are several CASE tools that supports change management. CASE tools are explained further in section 2.6.

All the above presented requirements engineering activities, and change management, are often combined and used together as requirements engineering processes. The following sections describe processes in general.

2.5 Requirements Engineering Processes

“Processes is an organized set of activities which transforms inputs to outputs” [Kotonya, Sommerville, 1997]. The purpose of processes is to help us perform work and projects successfully. Documenting the process used will help us repeat the success since it helps us remember how it was performed [Kotonya, Sommerville, 1997]. Another reason for documenting the process is to avoid failure and repeating mistakes. In requirements engineering several researchers have defined general processes that can be applied to companies with a need to improve their work with requirements, however many companies come up with their own models and processes or adjust the general ones to fit their organization. Three examples of requirements engineering processes are presented below.

2.5.1 A general requirements engineering process model

The model shown in figure 2.7 below is an example of a general requirements engineering process. It starts with a feasibility study where a decision is made whether to develop the proposed product or not. After the feasibility study the elicitation and analysis phase begins [Sommerville, 2001]. Next phase is to write a requirement specification. Often while doing this, new requirements are found and therefore there may be several iterations where the work alters between elicitation and specification writing. A similar iteration, between specification and validation, occurs when the validation phase starts. When all requirements in the specification has been validated a requirements document is produced.

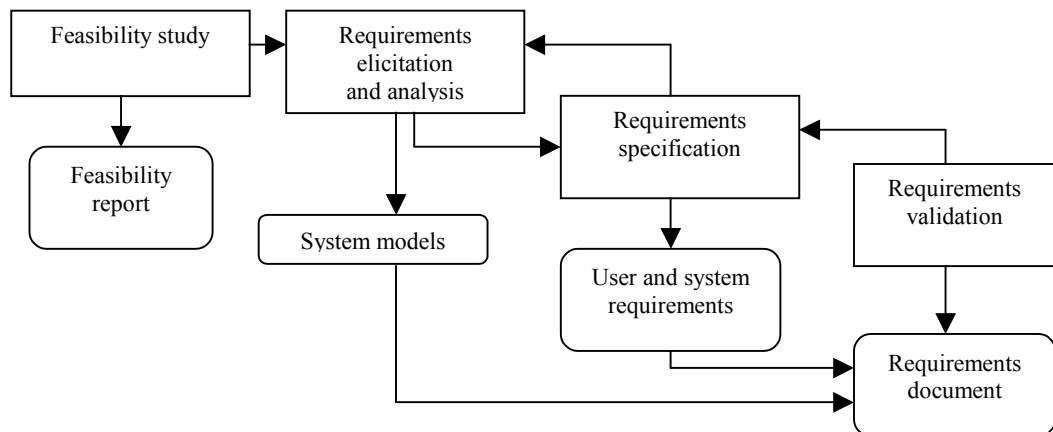


Figure 2.7 A general requirements engineering model [Sommerville, 2001]

2.5.2 The spiral model

Another development process model is the spiral model. The model in figure 2.8 shows the different activities of requirements engineering, in what order they are performed and that work can be remade until the requirements specification is of satisfaction for all stakeholders [Kotonya, Sommerville, 1997]. The spiral models characteristic is its iterative procedure. This is the main difference between this model and the above presented general model, which can be described as a waterfall model.

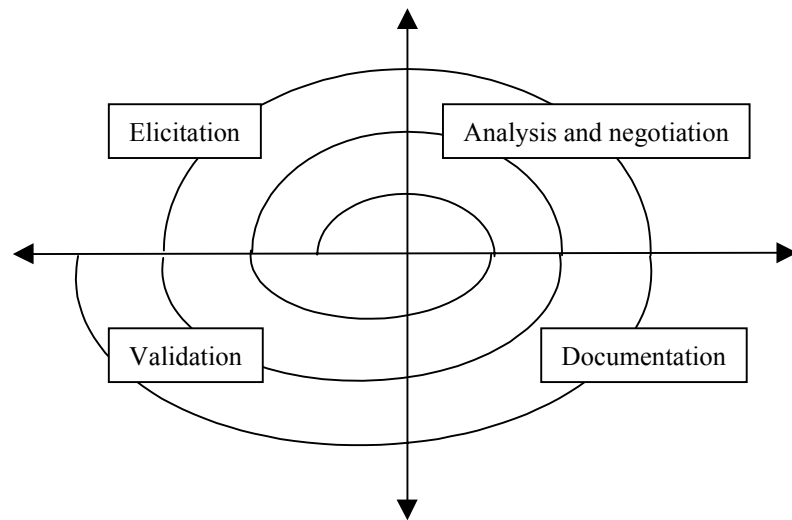


Figure 2.8 Spiral model

2.5.3 State-oriented model

REPEAT is an example of a requirement engineering process developed for a specific company [Regnell et al, 1998]. REPEAT stands for Requirements Engineering ProcEss At Telelogic. This process differs from the others since it applies a state-oriented life cycle showing the different conditions (states) a requirement can have. A requirement has to pass through different activities to get to a specific condition. This process is developed specifically for market-driven packaged software. Figure 2.9 shows the different conditions a requirement can have in the REPEAT 1.0 process.

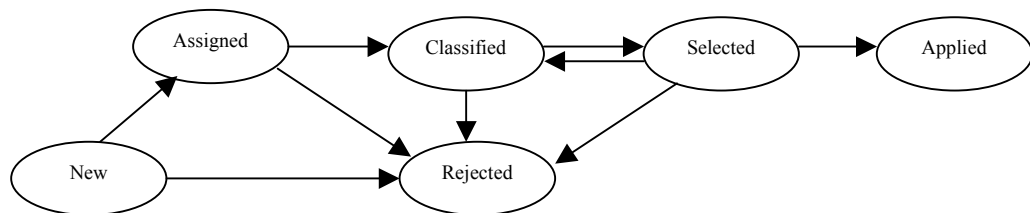


Figure 2.9 Requirement conditions in the REPEAT model

The conditions in figure 2.9 are further explained below.

New

The initial state of a requirement after it has been elicited and given an initial priority.

Assigned

The requirement has been given to an expert for classification.

Classified

The requirement is roughly estimated regarding cost and impact.

Rejected

End-state, for some reason the requirement will not be part of the requirement specification.

Selected

The requirement has been prioritized and selected for implementation.

Applied

End-state, the requirement has been implemented and verified.

Working with processes, activities and methods can be quite time-consuming. Using some kind of tool support may then be of great assistance.

2.6 Computer Aided Software Engineering tools

CASE tools (Computer Aided Software Engineering tools) are programs that support different software activities [Sommerville, 2001]. Today a wide range of tools is available that supports project management, configuration management, requirement analysis and change management, system modeling, testing etc. In large organizations where many projects are running in parallel, it is most useful to have CASE tools that support activities like these. According to [Kotonya, Sommerville, 1997] these tools may be divided in two types:

- Modeling and Validation tools
- Management tools

Modeling and validation tools support the development of system models used to specify the system and checking of models for completeness and consistency. Often these tools are modeling editors and checkers. Management tools help managing a requirement database and supports changes of requirements.

One area of requirements engineering where support is very limited is elicitation. This is difficult to support with general tools since the elicitation process and its stakeholders vary drastically between companies. Therefore most companies uses tools developed by themselves for their elicitation process support.

Examples of CASE tool that supports requirements engineering are Doors from Telelogic and RequisitePro from Rational.

3 Current situation analysis

In order to find out C Technologies current way of working, and possible future needs, we have performed a current situation analysis. This chapter describes the purpose, how the work was performed and the results from the analysis.

3.1 Survey

The current situation was performed as a survey where we aimed at finding out their current way of handling requirements. The survey was conducted with a questionnaire. The following sections explain how we performed the survey and its results.

3.1.1 Purpose

The purpose of the survey was to make an analysis of C Technologies current situation and in this way find future needs concerning their requirement process. It also covered how C Technologies is working with requirements today. This helped us gain a deeper understanding of the company and its processes in general. The different areas covered in the survey are:

- Stakeholders
- Elicitation
- Negotiation
- Prioritization
- Responsibilities
- Documentation
- Activities and methods

The survey includes gathering data about C Technologies requirements process and how its activities are performed. This helps us draw conclusions and obtain an understanding about which parts are performed well today and which need improvement. After having analyzed the data collected we should be able to propose requirements suitable for C Technologies' requirement process.

3.1.2 Performing the survey

A questionnaire was used to cover the main part of the data collection. The reason for choosing this approach was that it is easier and faster to analyze the result than if we had performed, for example, interviews. The first draft of the questionnaire was tested on an experienced project manager at C Technologies in the form of an interview. The purpose of the interview was to help us evaluate the questions, find further questions and setting up suitable answering alternatives for them.

Producing of questionnaire

To produce questionnaires and write questions, that are unambiguous and correct, is very difficult and time-consuming [Holme and Solvang, 1997]. First you have to specify the area on which the survey will focus and the result you are interested in.

From this you need to prepare questions that are correct and unambiguous. There are two main groups of questions, open-ended and closed-ended. Open-ended questions gives the respondent freedom to write answers in text, while closed-ended questions proposes alternatives that the respondent may choose from. Both alternatives are shown in figure 3.1 below. It is possible to merge the two alternatives by giving the respondent opportunity to motivate his/her answer on a closed-ended question. The analysis of closed-ended questions is much easier to perform than of open-ended because of the fact that it is hard to categorize and interpret the answers from open-ended questions.

Have you encountered any problems when working with high-level requirements?

Yes Partly No

Example of closed-ended question

Which problems have you encountered when working with high-level requirements?

Example of open-ended question

Figure 3.1. Examples of closed-ended and open-ended questions.

Using questionnaires involves some problems and risks. One problem is to motivate the respondent to take the time to complete the questionnaire and to actually consider each question carefully [Christensen, 1997]. If he/she does not, there is a major risk that the answers are wrong and that the result will be of no use. Another problem is that the respondents may interpret the questions differently and therefore give their answers out of different aspects. These risks shall be considered when analyzing the answers.

We designed the questionnaire iteratively according to the following method. We started by considering the results we were looking for and what information we wanted to extract from the answers. From answering these questions we could define the areas we wanted the investigation to focus on. The areas are listed in section 3.1.1 above.

Then we prepared questions about the issues relevant for our goal, which was to produce an improvement proposal for C Technologies high-level requirement process. When a first draft of the questionnaire was accomplished C Technologies' Usability Architect assisted us by reviewing the draft. We made recommended changes and

tested it on the project manager. His answers helped us with alternatives for the closed questions and gave us ideas about questions we had missed or forgotten. The questionnaire was revised one last time and then handed out to the selected respondents. The final version of the questionnaire is presented in Appendix B.

Selection of respondents

The selection of respondents is crucial for the outcome of the survey [Holme and Solvang, 1997]. Most relevant to us were the personnel at C Technologies that somehow work with requirements. Since we had no, or little, knowledge about who was working with requirements, C Technologies' Quality Director helped us with the selection.

All stakeholders were represented in the selection of respondents to the questionnaire. The stakeholders include personnel from all departments at C Technologies. In many projects C Technologies have an internal customer, who automatically gets included by choosing these respondents. The departments are:

- Development department
- Operations
- Management
- Marketing department
- Sales department

Including the end-users was not within the scope of the thesis since it would have required a separate questionnaire designed especially for them.

Twenty suitable respondents were found, and they were selected to participate in the survey. Since these persons were considered to have the best knowledge about the issue, we decided that this was many enough to gain an understanding of the current process and to determine the most important future needs. Had there been fewer suitable respondents this would still have been enough to perform the survey. To perform a scientific investigation with mathematical calculations to statistically determine the degree of truth of the answers obtained, at least thirty respondents would be required [Pagano, 1994]. This kind of investigation is more extensive and not necessary for our purpose.

The questionnaire was handed out, by us personally, to the respondents and they were given about four days to complete it. Of the twenty questionnaires handed out we received fourteen, which correspond to an answering rate of 70%.

3.1.3 Analysis of survey

The result from the analysis is the foundation for the "high-level requirement specification" produced in order to set demands on C Technologies requirement process. This specification later served as basis for our process improvement proposal.

Since much of our future work is based on this analysis it is important that the result is accurate and correct.

One way to analyze large amounts of data is to create an information matrix. With help from this matrix it is easy to produce other types of tables and matrices that simplify and clarify the information. They help us to determine tendencies and patterns within the information [Holme, Solvang, 1997]. When analyzing these matrices, the challenge is to determine if it is possible to extract such information.

Our analysis started with an information matrix where all answering alternatives were translated into numbers for easier measuring, see figure 3.2. From this, each question was analyzed without regarding any relations between them. The next step was to group relevant questions, search contradictions between them, or to try to strengthen the conclusions by finding similar answers. The questionnaire and the answers obtained are presented in appendix B.

Respondent	5	7	9	8	13	14	4	6	10	11	3	2	1
Position	1	1	1	2	3	3	4	4	4	4	7	8	9
Employment time	3	3	4	5	5	5	5	5	3	5	2	1	3
Q 1	2	1	2	1	4	2	2	4	3	2	2	2	2
Q 2	1	2	2	2		2	1	2	1	3	1	1	1
Q 3	1.2.4	1.2.3.4	1.2	1.2.3		1.2,3.4	1.3.4	3	1.2.3	1.2.3.4	1.2.3.4	1.2.3.4.5	1.2
Q 4	3	3	2	4	4	2	2	2	2	2	2	2	1
Q 4.2			2.3			1.2.5	1.2.3.4	1.3	3	2		1.2.3.4	1.2.3.5
Q 4.3	4	4	4					3					
Q 5		3	3	4		3	3	2	3	3	2	3	
Q 6	4	2		2	2	3	3	2	2	2	2	2	2
Q 7	2	2	1	2	3	3	2	2	2	2	2	3	2
Q 8	1	1	1	3	3	3	1	1	1	2	1	1	1
Q 8.1	1.2.3.4.5	1.2.4.5	1.2.3.4.5				1.2.3.4	1.2.3.4	1.2.3		1.2.3	1.2.3.4.5	1.2.3
Q 8.2						1				1			
Q 9	2	1	2	3	3	2	2	3	2	3	2	3	2
Q 9.1		2											
Q 10	4	4	4	4	4	4	4	4	4	4	3	4	4
Q 11	3	3	3	4		1	3	4	3	4	3	2	2
Q 12	1.2.5	1.2.4.5.6	2.3	2.5	1.2.5.6	2.3		1.2.3.5	1	1.2.4.5	1.2.4.5.6.7	1.2.3.4.5.6.7.8	2.3
Q 14	3	3	3	4	4	3	4	4	3	3	2	3	3
Q 14.2	4	1	1			1	1	1	1	1		1	1
Q 15	3	3	2	1	2	1	2	2	2	2	2	2	1
Q 15.2	3	4	3									3	3
Q 16	4	1	1	3	4	3	3	1	3	4		1	3
Q 18	6	3	1	5	6	1	1	1	1	6	1	6	2
Q 19	4	2	2	2	4	3	1	3	3	2	3	1	2
Q 20	2	1	2	4	2	1	2	4	1	2	1	1	2
Q 21	1	3	2	4	4	3	3	4	3	4	2	1	3
Q 22	2	2	1	3	3	2	2	2	2	2	2	1	2
Q 22.2	3	4				4	2	5	2	4	1	3	2
Q 23	1.3.4.7	1.2.3.8	1.4.7	1.3.4	1.3.4	1.3.4	2.4	1.3	1.3.4	1.3.4	3.8	1.2.3.4.5.6.7	1.2.3.4.7
Q 25	3	2	3	4	4	3	4	4	3	3	3	4	2
Q 27	3	3	3	4	4	3	4	3	3	3	3	3	3
Q 28	4		1.2	4	4	2	2		1	3	1.2	4	3
Q 29	3			3	3	1	2	1	3	1	3	1	1
Q 30	3	3	3	4	4	3	3	4	3	3	3	4	3
Q 31	3	1	1	4	4	1	2	1	1	1	2	3	1

Figure 3.2 Information matrix from our analysis

We used two variables to determine the credibility and relevance of the respondents' answers. These were "department" and "period of employment". C Technologies is a company under fast growth and almost every week new personnel are employed. Even if the most suitable respondents were selected we wanted to know their period of employment, which was considered when analyzing the answers. The reason for using "department" as a variable was to be able to decide on contradictions, disagreements or consent between departments.

Both variables are measured on the *nominal scale*. It means that the variable is divided into several categories, and the objects (respondents) are measured by determining the category to which they belong [Pagano, 1994]. Our categories for "department" are:

- Project manager
- Sales department
- Marketing department
- Development
- Patent department
- Support
- Management
- Purchasing department / Operations
- Other

For almost all of the questions an *ordinal scale* is used. This means that it is possible to determine if one object possesses more of a certain characteristic than another. Examples of the different scales we used are:

- Yes – No - Do not know
- Yes - Partly – No - Do not know
- Not at all – Little – Average – Much – Do not know

The following two sections present the result from our analysis. They are produced not only from our analysis but also with help from a CMM analysis carried out, by C Technologies' quality director, simultaneously with our survey. The quality director has interviewed personnel at C Technologies with the intent to establish C Technologies present CMM level, and also to determine on which areas to focus future improvement efforts. The results of the CMM investigation will be considered and used to support the conclusions made in our analysis. The CMM investigation is strictly internal at C Technologies and will therefore not be presented in this report. By controlling our result against the CMM analysis we have to some extent validated the result from our survey. The CMM analysis will also be of help producing the high-level requirement specification and when modeling and describing C Technologies current situation and future needs.

3.2 Result

Following sections present our main conclusions, a description of the current situation at C Technologies and their future needs.

3.2.1 Conclusions

Our main conclusions from the survey are:

1. No common known or documented processes are used
 2. Low utilization of proven elicitation and prioritization methods
 3. Responsibility for activities are not clearly documented
 4. Low degree of customer involvement in most requirement activities
1. On the question if “projects are planned according to specific work models or development processes” only 15% of the respondents answers “yes” and 62% answers “partly”. 8% answers “no” and 15% answers, “do not know”.

This indicates that they use some kind of process for accomplishing projects but the process varies from project to project. Our conclusion is that there is no uniform process used. Since the 62% of the respondents answer “partly” it is probable that the processes followed do not address all areas or activities needed. No defined process for how to elicit high-level requirements is used, but the respondents believe there is a need for one.

We have come to the conclusion that there is a need of processes, procedures and guidelines. The conclusion is based partly on the results from the analysis and partly from the literature we have read about requirements engineering, which suggests that it is important to work according to defined processes, see chapter 2.

2. When asking the respondents if “they use specific elicitation methods” 23% answers “yes”, and the only method mentioned is brainstorming. 62% answers “partly”, and 15% answers “no”.

When asking the same about prioritization only 15% answers “yes” and only one respondent of these can provide us with examples. 70% answers “no” and the remaining 15% answers “do not know”.

This clearly indicates that the utilization of proven methods, such as brainstorming, focus groups and pair-wise comparison, is very low.

3. When asking the respondents “who has the overall responsibility for requirements within a project”, the majority answers that it is the project manager. This is strengthened by the fact that when asked if “eliciting activities differ in different projects”, all respondents that have answered “yes”, say it is up to the project manager to decide method or procedure for the activities. Three project managers participated in the survey and for some reason they all have different answers to the question. This indicates that the common opinion is that the requirement activities are the project manager’s responsibility. This is however not documented.
4. Only 42% of the respondents are of the opinion that the customer is a stakeholder regarding high-level requirements, and just as many, say that the customer takes part in prioritization. The variation lies in the definition of the customer. There are internal customers and there are end-users. However, many respondents are of the opinion that the customer should be more involved. Involving the end-user is often very difficult when it comes to COTS products because of the fact that we often do not know who the end-user is. This might be one of the reasons for leaving them out.

3.2.2 Current situation at C Technologies

This is a description of the current situation at C Technologies based on the information derived from our survey, the CMM analysis, the reviewing of original source documents, further discussions with different stakeholders and C Technologies external and internal web sites.

Organization

C Technologies AB is a research and development company with head quarters located in Lund, Sweden. They also have one department located in Stockholm, where about 50 employees are working. C Technologies is the main company in a group of companies consisting of WeSpot, Anoto and C Technologies. The company has grown from around 50 to 250 employees in the last three years.

Most of their products are directed to customers like students and business-people. These categories of customers are believed to have a great need of simple and fast collection of information. More categories which might have this need are lawyers, journalists, teachers, hospital personnel and in some cases private persons.

Today C Technologies main product is C-Pen. C-Pen is a pen that reads, edits, translates and stores text. The latest release, C-Pen 800C, also includes a calendar, an address book, possibility to transmit text via mobile phones and larger memory capability than earlier models.

Mainly four types of development projects are performed at C Technologies:

- **Release projects** where updates and new releases of already existing products are developed.
- **OEM customizing projects** where specific customers need a special edition of an already existing product.
- **New development projects** are projects for development of products that are new to C Technologies and the market.
- **OEM technical projects** are projects where specific customers need to use C Technologies knowledge and techniques.

OEM is the department that handles customer specific projects and technical projects.

Between 1 and 10 development projects are normally in progress simultaneously, pre-studies excluded.

A project steering model, that is to be used in all projects, is currently under development.

Requirement engineering process

C Technologies have no documented process to follow when working with high-level requirements. Still elicitation, negotiation with customer, prioritization and sometimes documenting of requirements are performed. Today it is the project manager who decides the process and its activities and methods. Therefore most of the projects are performed with different process models that are not documented. Most of the respondents believe that having a process to follow would facilitate their work.

For the different types of projects there are different stakeholders regarding high-level requirements. The main stakeholders are development department, operations, marketing department, management and sometimes customers.

Some ideas about new products and new features for already existing products are gathered through an internal web-form and later evaluated. The evaluation is the basis for the decision, if the idea shall be further analyzed or not. Sometimes the high-level requirements are used for project planning, most frequently for time, cost and resource estimations. When a new project is decided on, it is the project manager who gets responsibility for the requirement activities. Other responsibilities regarding requirements are not defined.

When eliciting ideas no specific process is followed. The only method used is brainstorming. How it is performed varies between projects and depends on the project manager. The persons involved in the eliciting activity are the project manager and the persons he believes will contribute to the project. The end-user is rarely involved in this activity but sometimes, when customer specific products are produced, negotiations are held. The number of high-level requirements elicited for a product is estimated to be around 25, but this varies a lot.

The elicited requirements are prioritized, but no specific method is used. The marketing department together with management and development are usually the stakeholders involved in the prioritization. The time spent on prioritization is currently not measured. Prioritization is however thought of as an important activity. Requirements that are low prioritized are either moved to the next release or forgotten. Often low prioritized requirements that are excluded in the beginning of a project returns later with higher priority.

3.2.3 C Technologies future needs

Our opinion after the analysis is that C Technologies may need to improve or introduce the following:

- A defined and documented requirement process
Benefits:
 - All employees know how projects are performed.
 - All employees know what to do
 - All projects are performed in the same way
 - New employees can easier understand how projects are performed.
 - Better defined requirements
 - Better defined products
 - Some customers require that processes are used
- Methods for elicitation and prioritization
Benefits:
 - All involved stakeholders and actors' opinions are considered
 - Activities might be easier and faster accomplished
 - Better defined products
 - Facilitates the finding of the most important requirements
 - More ideas are gathered
- Definitions of stakeholders, actors and their responsibilities
Benefits:
 - Project members know whom to turn to with problems
 - All tasks get carried out
 - Increases the communication between the project members

4 Process Improvement Proposal

This chapter describes our process improvement proposal and how we developed it. The proposal aims at improving the requirements engineering process at C Technologies. Process improvement in general is often done by introducing phases, activities and methods that define the work to be performed [Sommerville, 2001]. We believe this approach is suitable for our proposal.

With chapter 3, and especially section 3.2.3, as basis we have defined goals for our proposal that especially addresses C Technologies current improvement areas and future needs. The goals we want to achieve with our process are:

1. To make all projects follow the same process
2. To make everyone aware of theirs and others responsibilities
3. To make the requirements better defined
4. To make the product definitions more detailed
5. To see to that all tasks get carried out
6. To make the elicitation more effective
7. To make sure that more ideas are gathered and documented
8. To make the prioritization more effective
9. To facilitate the finding of the most important requirements
10. To involve all stakeholders
11. To make sure that all ideas are evaluated

All goals aim at simplifying the requirements handling for all projects at C Technologies. In order to achieve these goals we will:

1. Introduce a specific requirements engineering processes
2. Try to increase the use of elicitation and prioritization methods
3. Define actors and their responsibilities
4. Try to involve all stakeholders
5. Make the process applicable to all C Technologies' technical projects
6. To make it easier and faster for new employees to join projects

In order to find relevant issues for our proposal we have studied general requirements engineering processes and models as presented in chapter 2. Suitable parts from existing models have been chosen and adjusted to fit C Technologies and their current way of working.

Especially one model has been studied in detail, the REPEAT model. In cooperation with the Department of Communication systems, Lund Institute of Technology Telelogic has developed this model. Trough articles [Höst et al, 2000], [Carlshamre, Regnell, 2000], [Regnell et al, 1998] and our supervisors from the Department of Communication systems, we have gained the knowledge needed about this model. The REPEAT model

was chosen since it has been used in practice and is proven to be suitable for development of market driven products. Research and evaluations with good result, has been made by Telelogic in cooperation with researchers at LTH.

When studying these models we have found general goals that most requirements engineering processes aim at fulfilling [Sommerville, 2001]. The goals are:

1. It shall help to improve product quality
2. It shall help to improve process quality
3. It shall facilitate the work with requirements
4. It shall increase customer satisfaction

We have also found several issues that a requirements engineering process contains. These will be considered further when developing the proposal. The issues are:

- Phases
- Activities
- How to store and document requirements
- Requirements attributes
- Requirements states (State oriented model, see chapter 2)
- How requirements are collected
- Actors participation and responsibilities
- Stakeholders participation and responsibilities
- Requirements estimations

The above stated issues and goals were used to produce a requirements specification describing an adapted process for C Technologies. Additional requirements were elicited from the results of the CMM analysis performed at C Technologies. The purpose of the requirements specification is to help us produce a process improvement proposal tailored for all C Technologies needs and wishes. The requirements specification is presented in appendix C.

From the requirements specification we have produced a process improvement proposal. At first an initial proposal was produced, which contained general phases, activities, stakeholders, actors, responsibilities and recommendations on how the activities can be performed. This proposal was reviewed and refined several times before the final version was accomplished.

4.1 Our Process proposal

This section defines and explains our Process Improvement Proposal. A reference version of the proposal is presented in appendix D. Our proposal has been developed to fit the project steering model at C Technologies as shown in figure 4.2. Each phase includes a number of activities, which will be the same for the different phases as shown in figure 4.1. All activities are not necessarily performed in each phase. The phases and activities are all explained in the following sections.

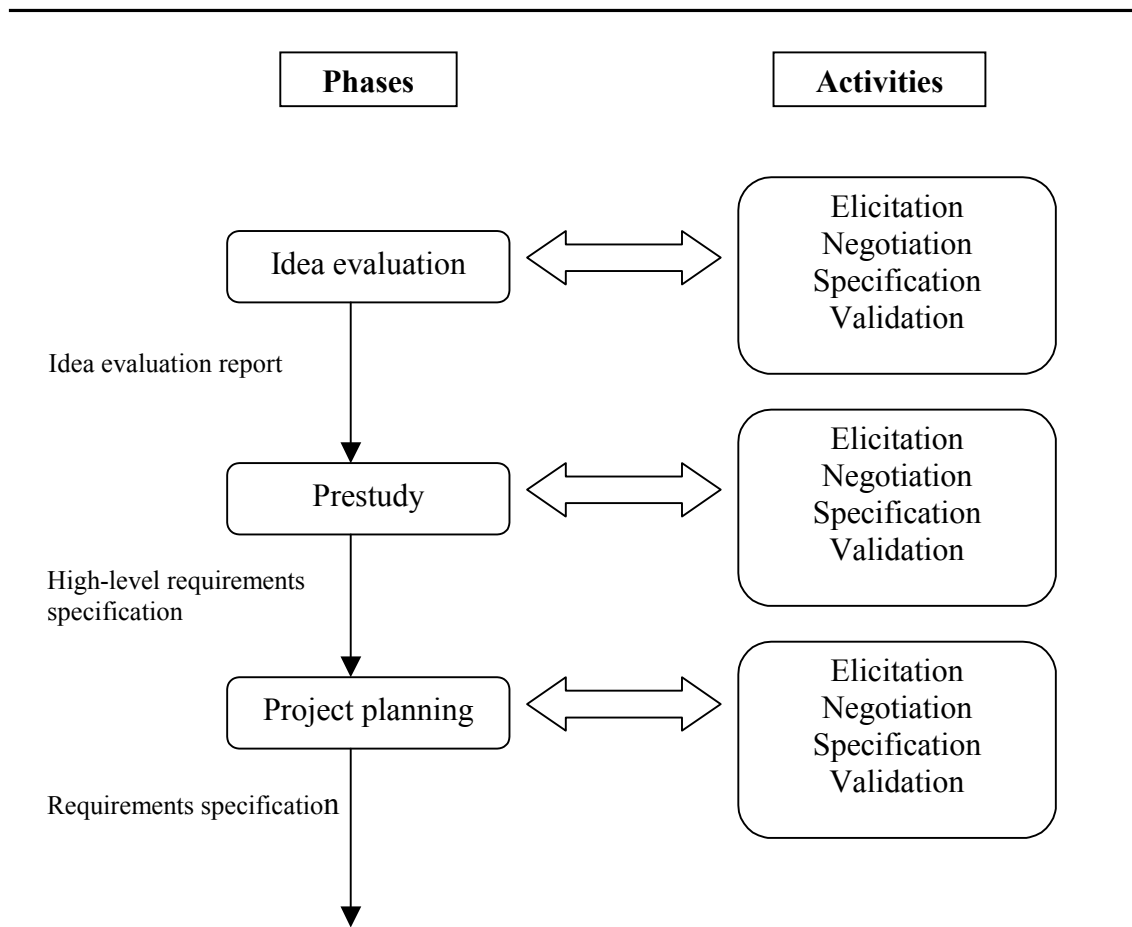


Figure 4.1. Mapping between phases and activities

Before the phases and activities are described we will introduce some actors who all will participate in the requirements engineering process in some way.

4.1.1 Actors and their responsibilities

All phases and their activities include different actors. The actors are persons who need to participate in the work. Every actor in the process has some kind of responsibility and it is important that everyone working with a project knows whom to turn to in different situations. The following are defined as actors:

Issuer

The issuer may be any person, at C Technologies or elsewhere, who comes up with an idea or a requirement. The issuers' responsibility is to submit a new idea or requirement with enough information to make further work possible. It may happen that the issuer needs to describe the requirement further on request from the requirements administrator.

Requirements administrator

The requirements administrator is responsible for new ideas and requirements getting estimated. The requirements administrator's main task is to find persons qualified to make the estimation and to put together the idea evaluation report. He/she also controls new ideas and requirements against old with the intention to find duplicates. He/she is thereby responsible for the very first selection of ideas that are to be further considered.

Product board

The product board is responsible for overall requirement management, prioritizing, decisions about selection or rejection for ideas and requirements before a project is decided on. The following roles shall be included in the product board:

- Development manager
- Innovation manager
- Quality manager
- Product managers: is responsible for meetings and arranged elicitations.
- Market representative

If a manager cannot participate he/she will send a representative who will take his/her place.

Project steering group

The project steering group shall support the Project manager who is responsible for overall requirement management, elicitation, prioritizing and decisions about selection or rejection for ideas and requirements after a project is decided on. The following roles shall be included in the project steering group:

- Project Manager: has the overall responsibility and is responsible for meetings and arranged elicitations.
- Management representative
- Customer: All projects shall have a defined customer. Internal projects will have a product manager who shall be considered as customer.
- Development representatives. Necessary for technical details.
- Quality manager

If a manager cannot participate he/she will send a representative who will take his/her place.

4.1.2 Process phases

Figure 4.2 below describes the three phases that constitute our proposal. These are compared to the early phases in C Technologies projectsteering model.

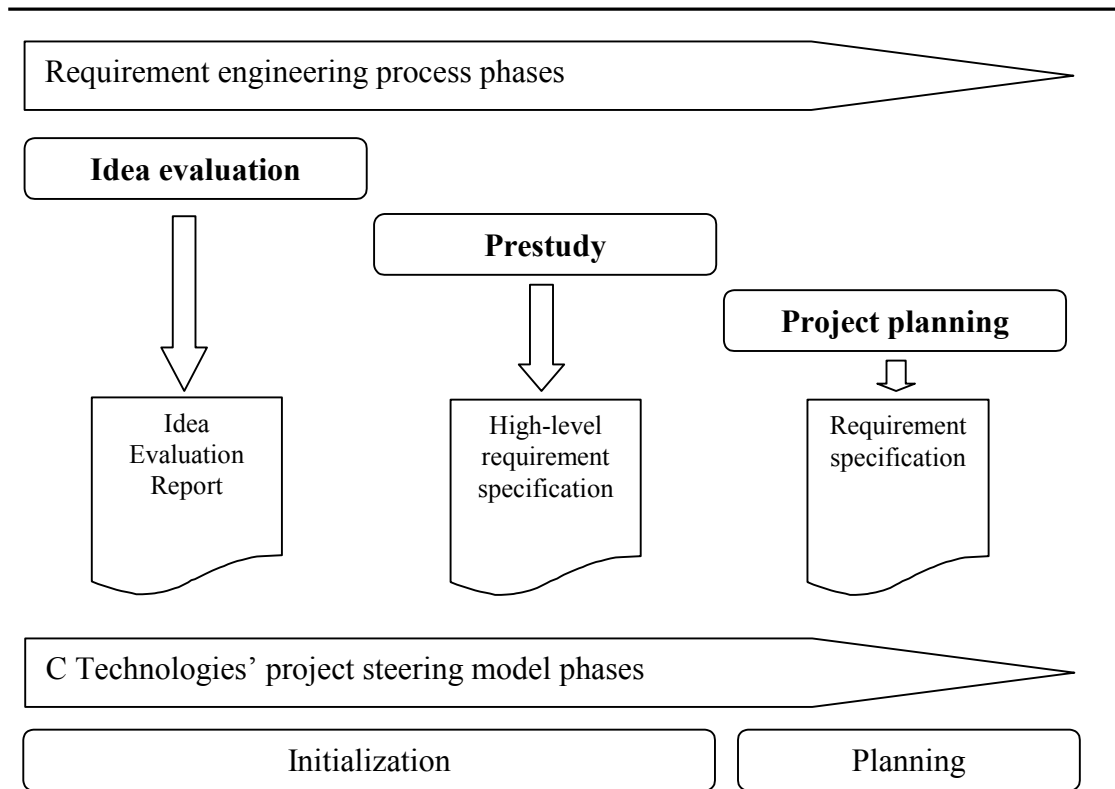


Figure 4.2. Requirement process phases and output

Idea evaluation phase

The purpose of the evaluation phase is to find, store, estimate and prioritize different ideas. This because decisions, about which ones are going further to a prestudy, shall be made. Today this is performed as an activity at C Technologies. Our proposal is to make it a phase that is included in the requirements engineering process.

Today all product development at C Technologies originates from ideas from employees at C Technologies, product users or customers requesting specific products. The ideas can be of various kinds, examples are:

1. They can propose completely new products or techniques.
2. They can propose new features for already existing products.
3. They can propose products or special features designed for a specific customer.
4. They can propose improvement projects, for example to lower manufacturing costs.

Every idea generated in the company must get documented, stored and evaluated. Ideas shall be collected continuously and whenever there is a need for new ideas a brainstorming or a focus group can be held. This could be if C Technologies decides to aim at a different customer group or considers developing new kinds of products. This we refer to as arranged elicitation. Both continuously and arranged elicitation is further described in section 4.1.4.

Estimation

The estimation in the evaluation phase shall be performed as follows. Each idea gets estimated out of four perspectives. The requirements administrator sees to that qualified employees make the estimations and the average value is put on an ordinal scale ranking from 1-8. At this level it is difficult to make exact estimates, which is why they are put on a scale. All estimates are collected and documented in an idea evaluation report. The estimations to be made are:

1. Technical value

How great is the technical value considering the following aspects?

- New technology
- Core technology
- Patent issues
- Etc

2. Market value

How great is the market value considering the following aspects?

- Market attraction
- Market demand
- Scalability
- Etc

3. User benefit

How great is the user benefit considering the following aspects?

- Usefulness for the user
- Similar products from competitors
- Etc

4. Development effort

How great is the development effort considering the following aspects?

- Time
- Cost
- Resources
- Realization possibilities
- Etc

The estimates defined above are chosen to make it easier to prioritize the ideas. Since they give the persons who are prioritizing useful information they are more likely able to make a correct decision. On a regularly basis a number of completed idea evaluation reports shall be prioritized and decided about by the product board. The intention is, that every idea that is accepted by the product board shall lead to a prestudy or be part of a project. Maybe several ideas are accepted but there are not enough resources to make prestudies of them all. Here it is possible to prioritize the ideas. It is also possible that many ideas are comprised into one project.

Prestudy phase

The purpose of the prestudy is to investigate ideas further to be able to decide whether to develop the proposed product or feature. This is done both from a technical and market point of view. The input to the prestudy is the idea evaluation reports produced in the idea evaluation phase. In cases when the idea is about features to an already existing project it may be unnecessary to perform a prestudy. In this case the idea can be put into an ongoing project as a change request, or in a project that is in its planning phase.

In the beginning of the prestudy it is possible to elicit more ideas and requirements for the product, or breaking down requirements into more detailed ones. By conducting a brainstorming or a focus group, see section 2.4.2, this can be accomplished.

Estimation

All requirements found shall then be estimated. As in the idea evaluation phase, qualified employees will make the estimates. In this phase the estimates will be more detailed and every estimate will consist of three values.

1. Minimum estimate – the smallest estimate
2. Probable estimate – most probable outcome
3. Maximum estimate – the largest estimate

C Technologies project steering model defines several perspectives that shall be considered for the collection of requirements belonging to a prestudy. The perspectives are technical possibilities, qualification and resource analysis, financial analysis, SWOT analysis, patent and production. Making cost estimations on each requirement facilitates the financial analysis. No other perspectives are relevant to consider for each requirement. They are only for entire projects. This leads to that each requirement shall be estimated out of the following aspects:

- 1. Development cost**

Development cost for the requirement shall be estimated and given in Swedish kronor.

- 2. Manufacturing cost (per unit)**

Manufacturing cost for the requirement shall be estimated and given in Swedish kronor.

As in the idea evaluation phase these estimates shall facilitate the prioritization and the decision-making. The estimates are chosen since they will represent the entire cost for a requirement. After all requirements, for a specific product, are estimated it will be possible to find the break-even point that describes what price a product must have in order to bring profit.

The prestudy shall result in a high-level requirements specification containing all requirements with their detailed level estimates. From the information in the high-level requirements specification the product board shall be able to decide whether to run a development project for the proposed product or feature. It shall also be decided if all requirements from the prestudy or only a selection of them shall be input to the project planning phase. In cases when more than one prestudy has been performed, and decided on, it may be necessary to prioritize them.

Project planning phase

At this point it has been decided to run a project for a certain product or add a feature to an already existing product. An idea evaluation report or high-level requirements specification will be input depending on what kind of project it is. Most probably more requirements need to be elicited, requirements from earlier phases need to be described in more detail or broken down into more detailed ones. In some cases it will be unnecessary and even impossible to use all requirements from the high-level requirements specification, instead a limited set of the high-level requirements will be chosen for further work. As in the earlier phases more requirements can be elicited by conducting a brainstorming or focus group, see section 2.4.2.

Estimation

In this phase as well as in the others, estimations are to be made. This phase uses the same method as in the prestudy phase where each requirement gets three values. The reason for making additional estimations in this phase is that the requirements might have been described in more detail. That is why the level of detail for the estimates also has increased. The estimates, which shall be made here, are cost and time estimates. These will hopefully make the planning of the entire project easier and more accurate. The estimates are:

1. Development time

The development time for the requirement shall be estimated and given in days.

2. Hardware development costs

Hardware development cost for the requirement shall be estimated and given in Swedish kronor. All tools and component costs, and time shall be considered.

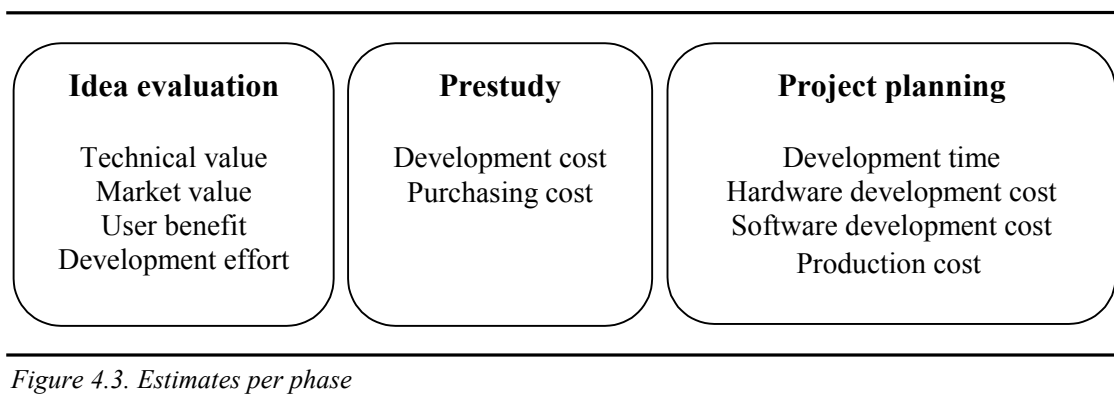
3. Software development costs

Software development costs for the requirement shall be estimated and given in Swedish kronor. Computer costs, development programs costs and time shall be considered.

4. Production costs

Production cost for the requirement shall be estimated and given in Swedish kronor.

The Project planning phase shall result in a requirements specification containing all requirements and their estimations. The requirements specification shall in this way define the product to be developed and also help the project manager to plan the project. Figure 4.3 below summarizes which estimations that are performed on a requirement in each phase.



4.1.3 Requirement states

Through its lifecycle every requirement passes different states. The reason for having the states is that it shall be possible to track requirements out of different aspects. Both Telelogic's REPEAT and Ericsson's RDEM models [Carlshamre, Regnell, 2000] have used this approach with success.

1. It is possible to see how far a requirement has come in its lifecycle.
2. It is possible to determine the information content of a requirement.

The different states a requirement can have is presented below and shown in figure 4.4. The arrows describe the normal flow for a requirement.

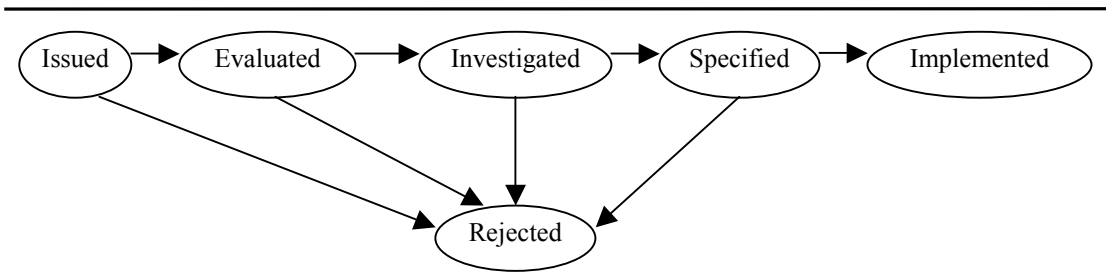


Figure 4.4. Requirements states in normal flow

Issued

In order to enter the process, an idea or requirement must be documented. What it enters the process it is assigned the state issued. Breaking down a requirement into more detailed ones leads to more issued requirements. All issued requirements are input to the Idea evaluation phase. The requirements administrator is responsible for transferring an issued requirement into another state. An issued requirement must at least contain the following attributes. The attributes are further explained in section 4.1.5.

1. Id
2. Date
3. Title
4. Description
5. Type
6. Issuer

Evaluated

An evaluated idea shall contain all attributes from the idea evaluation phase. Based on this information it shall be decided if the requirements shall move on to the prestudy phase. An evaluated requirement is usually input to the prestudy phase and when it is decided on it is considered as a requirement. An evaluated requirement must at least contain the additional attributes of the specific estimates for the idea evaluation phase as shown in figure 4.2:

1. Technical value
2. Market value
3. User benefit
4. Development effort

Investigated

An investigated requirement shall contain all attributes from the prestudy phase. Based on this information it shall be decided if the requirements shall move on to the project planning phase. An investigated requirement is usually input to the project planning phase and when it is decided on the intention is that it shall be part of the final

requirement specification. An investigated requirement must at least contain the additional attributes of the specific estimates for the prestudy phase as shown in figure 4.2:

1. Developing cost
2. Manufacturing cost

Specified

A specified requirement is in adequate detail and consists of all needed estimates. It is also ready to be broken down into project activities. A specified requirement must at least contain the additional attributes of the specific estimates for the project planning phase as shown in figure 4.2:

1. Development time
2. Hardware development costs
3. Software development costs
4. Production costs

Implemented

As soon as a requirement has been fully implemented and verified by the customer it shall be moved into the implemented state. This makes it possible to find earlier implemented requirements, which may lead to more reuse of already implemented features and functions, which may decrease development time and costs. This is an end state for a requirement.

Rejected

An idea or requirement that is rejected shall lead to no further work. Several things can lead to the rejection of a requirement. It may be a duplicate, already implemented, a bad idea or because it was not compliant with the company strategy. When a requirement gets rejected a comment about why it was rejected shall be added. This is an end state for a requirement.

The diagram in figure 4.5 illustrates how the phases of the process interact with the requirement states. During its way through the process a requirement can pass three phases and four states unless it is rejected. As shown in the diagram, the level of information contained in a requirement increases for each phase it passes.

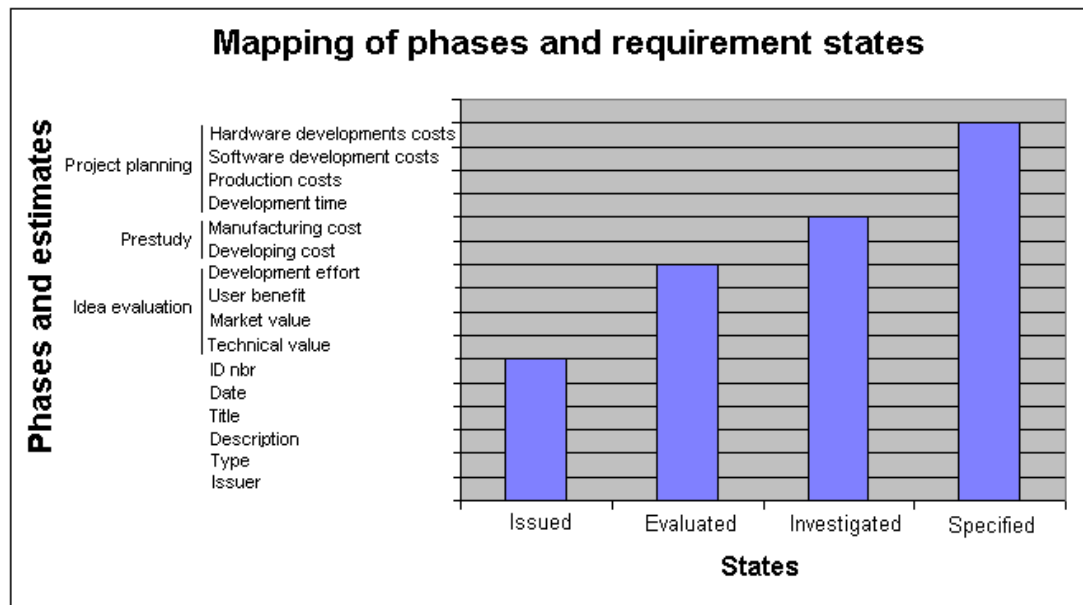


Figure 4.5 Mapping between phases, states and estimates

As described in scenario 2, section 5.2, not all requirements pass through all phases and all states. These requirements do not get all estimates. The reason for making the estimations is to facilitate the prioritization and decision-making in a specific phase, and if a requirement skips one of the phases it is unnecessary to perform the estimations for that phase.

4.1.4 Process activities

Within each phase a number of activities take place, some of them has been mentioned in the description of the phases but will be further explained here. The activities are elicitation, negotiation, specification and validation. Elicitation is divided into arranged and continuous elicitation and documentation and negotiation is divided into estimation and prioritization as presented in figure 4.6.

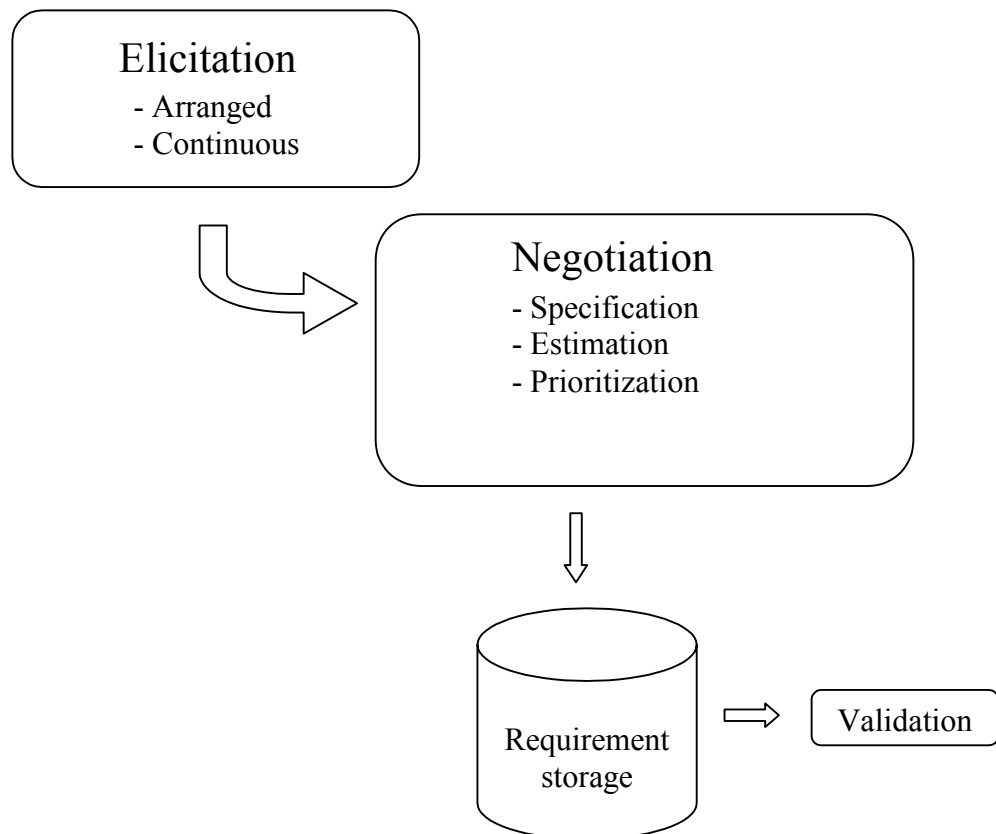


Figure 4.6. Requirement activities

The activities are applicable to *each* phase in our proposal, see section 4.1.2, with just minor modifications. A requirement may iterate between the activities if its information or formulation somehow needs to be changed.

Elicitation

This activity is divided into arranged and continuous elicitation and documentation. The purpose of the elicitation activity is to find and collect ideas and requirements, by involving all stakeholders, for the current project. The reason for dividing elicitation into arranged and continuous is to be able to capture an idea or requirement the minute it appears, as well as it shall be possible to extract new requirements from an earlier idea or requirement.

Arranged elicitation

There are several reasons for performing an arranged elicitation:

- To find more requirements on a product or feature.
- To find new product concepts.
- To break down high-level requirements into more detailed ones.

Two alternatives for arranged elicitation is to conduct a brainstorming or a focus group, which are both described in section 2.3.2.

Several stakeholders must be included when performing an arranged elicitation. The stakeholders are:

Customer

The customer must be present/represented at all arranged elicitation. The customer can be internal or external. One or more representatives from marketing department or the product manager are likely to be internal customers. External customers may be representatives from other companies interested in ordering an especially for them developed product.

Support

The personnel working with support at C Technologies have more contact with the consumers than anybody else. Therefore it is important to get their opinions about new products to be developed.

Development

Personnel from the development department are the ones who will be affected by the elicited requirements since it is they who will actually use the requirements later in the project. They also provide input on technical possibilities that might lead to new ideas. It is therefore important that they get to share their opinions and technical expertise.

Project management

They are supposed to be in overall control of projects and shall therefore be present and take part of the elicitation.

Continuous elicitation

Continuous elicitation is the spontaneous collection of ideas and requirements. The spontaneous collection is made through a web-form where all stakeholders, employees and end-users are able to submit their ideas. Everybody mentioned must know of this possibility. The ideas and requirements will be submitted directly into the process. The submitters (issuers) are:

Customers and end users

Customers and end users shall be able to submit ideas and requirements through an external web form or by talking to personnel at C Technologies i.e. market representative, support or other.

Employees at C Technologies

All personnel at C Technologies shall be able to submit their ideas and requirements through an internal web form. This can be made through an internal web-form or participation in an arranged elicitation.

Negotiation

This activity is divided into estimation and prioritization. It also contains the decision-making. After all requirements are prioritized a decision has to be taken.

Specification

All requirements that get elicited must be documented. The process proposes that all ideas and requirements shall be stored in a requirements database. This means that whenever an arranged elicitation has been performed, one of the participants gets the assignment to submit the elicited ideas to the database. The ideas that are continuously elicited through the web-form are submitted straight into the database. The web-form sees to it that all the information needed are entered. Additional benefits of using a database are presented in section 5.1.

All three phases, presented in section 4.1.2, result in some kind of specification. The specifications from idea evaluation and prestudy shall be used as foundation for decision-making, and the specification from project planning is the requirement specification for the project.

Estimation

The purpose of estimating requirements is to extract useful information about the product. Having information about the requirements such as probable development time, cost, user benefit, salability etc. makes it easier to prioritize and make decisions about the requirements. The estimates will also be of help when planning and making budgets for projects. Each requirement is to be estimated. In section 4.1.2 the estimates to be made for each phase are presented.

There are different stakeholders regarding estimation. For every project the main responsibility shall be defined. One person will be requirements administrator and his/her task is to make sure all requirements get estimated. This means that for every requirement the requirements administrator must find persons suitable to make the estimations. These persons will probably be found among development personnel, marketing personnel or management.

Prioritization

It is very common, that more requirements than are possible to realize are elicited, see section 2.3.5. It will most likely be necessary to prioritize among them. One method that is suitable for prioritizing requirements is a cost-value based method with pair-wise comparison [Karlsson, 1996]. The method enables comparing requirements against each other where a specific criterion is considered. It is possible for the user to choose criterion. Criterion that shall be used is the estimates made for the requirements.

After prioritizing the requirements with pair-wise comparison they shall be assigned a level of importance. The levels are:

- High Must be implemented
- Medium Implemented if time
- Low Next release or other product

One CASE tool that supports pair-wise comparison is Focal Point. It has several suitable features to perform prioritization. Focal Point provides a decision support portal in which prioritization of requirements can be made [Focal Point, 2001]. Focal point makes it possible to perform the following:

- Define evaluation criteria such as Value for customer, Time for implementation and Salability for prioritization purposes. The user chooses the criterion.
- Prioritize the requirements according to the evaluation criteria using smart pair-wise comparisons, i.e., determine which of two requirements that best fulfill a criterion and to what extent.
- Prioritize the requirements individually, in group-session, or over the Web.
- Check the consistency of the pair-wise comparisons in order to identify and resolve contradictions.
- Reprioritize the requirements continuously.

Validation

The validation activity is about determining that the requirements specification produced in the project planning phase fulfills a number of quality attributes. Validating the specification is important since it will lead to less rework and less changes. The major purpose is to make sure it satisfies the customer's demands, wishes and needs, and that it really defines the product to be developed. The validation is made through a review where the specification is checked against a list. The project steering group is responsible for the review. A review checklist example is presented in appendix E.

4.1.5 Requirement attributes

To support our proposal a requirement shall be able to contain at least the attributes presented below. These attributes are related to the earlier defined states and phases.

- Unique identifier (number)
- Title
- Description
- Issuer email
- Issuing date
- Type (functional, nonfunctional, usability)
- State (issued, evaluated, investigated, specified, implemented, rejected)
- Estimates (technical value, market value, user benefit, development effort, development cost, manufacturing cost, development time, Hardware development costs, Software development costs, production costs)
- Estimate outcome (makes it possible to compare the estimate and its outcome, which is necessary for process improvement)
- Priority (high, medium, low)
- Expire date
- Additional comments

The identifier makes the requirement unique, this helps finding a specific requirement. Title and description defines the requirement, describes the idea or requirement. Without these attributes the rest is of no use. The issuer email makes it possible to locate and contact the issuer. This might be needed for further explanation. The date tells when the requirement was issued and the type is used to group requirements. Grouping of requirements is described in section 2.3. The state shows the information content and where in the lifecycle a requirement is, as explained in section 4.1.3. the estimates are several attributes which are made on each requirement to ease the planning and decision making for projects. Different attributes belong to different phases according to section 4.1.2. The priority shows the importance for a requirement, for example if it must be implemented or not. Expire date is used when the requirement is time critical. That is if the requirement must be handled before a certain date to be useful. If it is not handled before this date it may be automatically rejected. This function sees to that old, not handled, requirements leaves the process. It shall also be possible to make additional comments for a requirement. These might be about why a requirement shall be implemented, how it might be implemented or why it was rejected.

5 Process realization

Our process proposal contains phases and activities that must be executed in every project, that is, over and over again many times. To be able to work according the proposed process there are tools, documents and other facilities that must be in place first. Examples of these are web forms for collection of ideas and requirements, document frameworks, a database for storage of requirements and tools that supports different activities.

We recommend the using of a database for storage of requirements. As we see it today there exists three ways of storing requirements. The first is to keep all requirements in the head, to remember them. The second one is to write them down on paper and the third one is to store them in a database. As we will try to show the database provides several possibilities that the first two cannot manage.

We have prototyped a web-based requirements storage tool using a database with the intension to explain the advantages and possibilities with this kind of requirements storage. The prototype shall be considered as a throwaway prototype. This means that it only is an example that is used to express the functions we believe relevant as a first step towards a more structured requirements engineering process. If C Technologies decides to work with a requirements database, and they want to implement their own, they have the possibility to use our prototype to elicit requirements for the new one. Another possibility is to purchase a suitable requirements CASE tool. Examples of existing CASE tools, which include a requirements database, are DOORS by Telelogic and RequisitePro from Rational.

In section 5.2, three scenarios are presented. They further describe how requirements should be handled according to our process proposal together with the prototyped database tool.

5.1 Database prototyping

Our intension with the database prototype is to show how it may facilitate and simplify the use of our process proposal and to show how the process will work in “real life” situations. That is, the benefits from using a database to store requirements. This is the only purpose of the prototype and therefore the aspects of performance, usability, interface etc, has not been considered further.

The following are considered benefits of using a database:

1. Simplifies the use of the process
2. Stores all requirements for as long as we want
3. Makes it easier to produce a specification (outputs)
4. Makes tracing of requirements easier
5. Stores the history for a requirement, that is, all updates and modifications

6. Contains storage for continuous elicitation as well as for arranged
7. Makes it possible to easy modify requirements
8. Supports easy requirements access for all stakeholders
9. Supports grouping of requirements, for example functional – non functional
10. Makes it possible to define different permissions for specific users, that is to define who shall be able to change the database

As mentioned earlier, the purpose of our proposal is to facilitate the work with requirements. Our proposal is developed with this in mind and using a database, with the following functionality will support our proposal.

The following are recommendations of what a requirement storage tool shall be able to do, to support our proposal:

1. Submit a new requirement
2. Store requirements
3. Find requirements through search functions
4. List all requirements with regard to specific criteria, for example:
 - List all requirements with a specific state
 - List all requirements belonging to a specific project/product
 - List all requirements issued by selected issuer
5. Break down a requirement into more detailed requirements
6. Modify and update requirements, for example, change its state or add a new estimate
7. Support the producing of specification outputs
8. Find the origin of a requirement
9. See the history of a requirement (the history is the changes that have been made to a requirement)
10. Extract the estimates on a requirement
11. Extract what state and/or priority the requirements has
12. Find the issuer for requirements
13. Merge two requirements into one
14. Support prioritization
15. Force the issuer to enter the required information
16. Support permission definitions for different users (actors)

Using databases for storing of requirements also involves problems. A problem may occur when a new requirement is added which could be split into already existing requirements. Another problem is that several requirements could be similar or even identical [Natt och Dag et al, 2001]. This due to the fact that many requirements are handled and several stakeholders are involved. Finding these identical requirements is necessary because we do not want to handle the same requirement twice. Requirements might also be in the database too long before being taken care of, and the administration of the requirements too extensive. These problems are not specifically for database use but also appear when using other methods for storing requirements. The advantage with the database is that it is possible to give tool support to handle these problems.

5.1.1 The actual database prototype

Our prototype contains a selection of the most important features for us to be able to show what happens to a requirement from the moment it has been issued until it is implemented. The features implemented in our prototype are:

- Submitting of requirements
- List all requirements
- List requirements belonging to specific project
- List requirements belonging to specific project and specific state
- Break down requirements into more detail
- Updating requirements (all attributes possible to update and edit)
- View the history of a selected requirement

For readers interested in how the database and the web forms are implemented we present this information in appendix F and appendix G.

Further follows pictures of the web tool developed and explanations for each function. The first picture (figure 5.1) shows the form where ideas and requirements are first submitted. The intention is that all ideas shall be submitted through this form into the database. This way all ideas get stored and later evaluated. The attributes that have to be submitted are title, description, type, state and the submitters (issuers) email. It is also possible to write additional comments for the idea.

Requirement Collection Form (prototype)

Product: What product does this requirement mainly affect?

Title: A one line title that describes the requirement

Description: A detailed description of the requirement

Type: Declare what type this requirement is

Submitters e-mail: Add submitters (your) e-mail

Comments: Any other comments you want to make in order to promote this requirement. For example why this requirement should be implemented or what problem this requirement is trying to solve.

[Submit a requirement or an idea](#)
[Search function](#) (Update, Break down or View history for selected requirement)
[Requirement Service start point](#)

Prototype produced by [Urban Martinsson](#) and [Åsa Karlsson](#) as part of our Bachelor Thesis
Last updated 2001-06-14

Figure 5.1. Submitting new idea

Figure 5.2 shows a list of requirements belonging to a specific project. In the prototype it is possible to list all requirements in the database, list requirements for different projects or list requirements with a specific state for a selected project. This function shows all attributes belonging to a specific requirement. For real use it would be preferable if this function made it possible to list requirements with different states, of different types etc, some kind of search function. It would also be preferable if the attributes were selectable. It might not always be wanted that all attributes are displayed. The function of listing requirements are found under a hyperlink called search, which is present at each page.

Id	Title	Description	Type	State	Cost estimate	Technique level	Salability	Customer need	Priority	Parent
47	Minneskapacitet	Minneskapaciteten för C-Adress ska vara minst 250 adresser	Non-functional	Issued						
48	Typsnitt	Storleken på typsnitt ska ligga mellan 5-22 punkter	Non-functional	Issued						
49	Menyspråk	Menyspråk och OCR-språk ska vara på köparens lands modersmål	Non-functional	Investigated	5	5	8	8	3	
50	Typsnitt	Storleken på typsnitt skall vara 18 punkter	Non-functional	Issued						48

Additional comments	Date	Issuer Email	Update	Break down	View history
	2001-05-31	edictionary4@cpen.com	Update	Break down	View history
	2001-05-31	edictionary4@cpen.com	Update	Break down	View history
	2001-06-07	urban@cpen.com	Update	Break down	View history
	2001-05-31	edictionary4@cpen.com	Update	Break down	View history

Figure 5.2. List requirements for specific project

As the picture shows, at the end of each requirement there are three functions (hyperlinks). That is, three things that is possible to do with each requirement. The first function is *update*. With this function it is possible to update the requirement attributes, as shown in picture 5.3. It is possible to update all attributes from the submitter form, except the project to which the requirement belongs. It is also possible to add new information such as different estimates and a priority for the requirement. The project displayed as the first attribute in figure 5.3, is a list of projects. It is possible that one requirement belongs to more than one project.

Requirement Collection Form - Microsoft Internet Explorer

Address: <http://localhost/modification.asp?requirement=47>

Project:	boda	The projects this requirement is connected to
Id:	47	This requirement identifier
Title:	<input type="text" value="Minneskapacitet"/>	If necessary, specify the title more
Description:	<div><div>Minneskapaciteten för C-Adress ska vara minst 250 adresser</div></div>	If necessary, specify the description more
Type:	<input type="text" value="Non-functional"/>	Declare what type this requirement is
State:	<input type="text" value="Issued"/>	Declare what state this requirement has
Development effort:	<input type="text"/>	Specify the development effort by selecting a number
Technical value:	<input type="text"/>	Specify the technical value by selecting a number
Market value:	<input type="text"/>	Specify the market value by selecting a number
User benefit:	<input type="text"/>	Specify the user benefit by selecting a number
Priority:	<input type="text"/>	Declare what priority the requirement has
Parent:		The requirement this requirement originate from
Original issuing date:	2001-05-31	The date when this requirement was last updated
Editors e-mail:	<input type="text" value="cdictionary4@open.com"/>	Update this field with the editors (your) e-mail
Comments:	<div><div></div></div>	Any other comments you want to make in order to promote this requirement. For example why this requirement should be implemented or what problem this requirement is trying to solve.

Figure 5.3. Update requirements

The second function for a requirement, as shown in figure 5.2, is *break down*. Breaking down a requirement means that it is possible to create additional requirements with the selected requirement as parent. That requirement becomes the main requirement and the new ones are connected to that. This function shall be used when a requirement is not detailed enough. With this function, new more detailed requirements are then created see figure 5.4.

Requirement Break down Form

Product: bodn
 Father requirement: 47
 Title: Minneskapacitet

Description:

Type: Non-functional
 State: Issued

Submitters e-mail:

Comments:

Submit requirement Reset

The projects this requirement is connected to
 The requirement this requirement originate from
 The title of the parent requirement, and this new one

A detailed description of the requirement

Declare what type this requirement is
 Declare what state this requirement has
 Add submitters (your) e-mail

Any other comments you want to make in order to promote this requirement. For example why this requirement should be implemented or what problem this requirement is trying to solve.

- [Submit a requirement or an idea](#)
- [Search function](#) (Update, Break down or View history for selected requirement)
- [Requirement Service start point](#)

Prototype produced by [Urban Mattsson](#) and [Åsa Karlsson](#) as part of our Bachelor Thesis
 Last updated 2001-06-14

Figure 5.4 Break down form

The third function for a requirement, as shown in figure 5.2, is *view history*. Every time a requirement is updated the earlier information stored in each attribute is saved. It is possible that a requirements content changes and that it no longer describes the same function as it was intended to do. The *view history* function then provides the possibility of finding out who made the critical change to the requirement and when it was made. This function also makes it possible to decide different durations that a requirement has for example a specific state, how often it is updated etc. This information might be useful in the future when changing or updating the entire process. The view for this function is shown in figure 5.5. The gray shaded area shows the requirements actual status.

Id	Title	Description	Type	State	Cost estimate	Technique level	Salability	Customer need	Priority
49	Menyspråk	Menyspråk och OCR-språk ska vara på köparens lands modersmål	Non-functional	Issued	0	0	0	0	0
49	Menyspråk	Menyspråk och OCR-språk ska vara på köparens lands modersmål	Non-functional	Issued	5	5	8	8	0
49	Menyspråk	Menyspråk och OCR-språk ska vara på köparens lands modersmål	Non-functional	Investigated	5	5	8	8	1
49	Menyspråk	Menyspråk och OCR-språk ska vara på köparens lands modersmål	Non-functional	Investigated	5	5	8	8	3

Additional comments	Date	Issuer Email
	2001-05-31	cdictionary4@cpen.com
	2001-05-31	cdictionary4@cpen.com
	2001-05-31	urban@cpen.com
	2001-06-07	urban@cpen.com

Figure 5.5. View the history for selected requirement

As stated earlier the intention of the prototype is to help us explain a requirement passing through the different phases and activities from the moment it has been issued until it is implemented. Therefore we have chosen to implement the, above, presented functions. A tool that is to be used in reality must contain more functions, some of them also presented in the above sections.

5.2 Requirement Scenarios

To further explain how the process proposal works, three scenarios have been written. The three scenarios are walk-troughs where an idea passes the process. It is impossible to write scenarios that cover all possible events. Therefore we have chosen to present, what we believe, the most describing events. The scenarios are also illustrated with figures.

1. The first scenario describes what happens to an idea that suggests a completely new product.

2. The second scenario describes what happens to an idea that suggests a new function for a product, which has already entered the project planning phase. That is, after it is decided that a project shall start.
3. The third scenario describes what happens when ideas about a new release of an existing product continuously are issued. The scenarios are presented below.

1. Idea for a new product

Someone comes up with an idea for a completely new product and submits it into the database. He/she becomes the issuer. The requirements administrator gets a notification by email. The state of the requirement is automatically put to issued.

Idea evaluation phase

The requirements administrator, who is responsible for the idea evaluation looks in the database, at regular intervals, for new ideas. He/she then finds the issued idea. He/she then looks up persons that are suitable to make the estimates in this phase. He/she puts together their opinions and updates the idea in the database with the estimates, and updates its state to evaluated. With this information he/she produces an idea evaluation report.

The next step is to present the idea evaluation report to the product board. The product board has two choices, either to approve the idea or reject it. An approved idea shall be passed on to the next phase, and is thereby treated as a requirement. A rejected requirement gets the state rejected, and is no longer considered. It is the intention that all approved ideas shall continue through the process, but it is possible that the product board must decide about several ideas for new products and maybe there are not enough resources to perform prestudies of them all. In this case it may be necessary to prioritize the ideas. It is also possible that several ideas get comprised into one product. All these ideas, now treated as requirements, are input to the prestudy phase.

Prestudy

After an idea has been approved a project manager gets the responsibility for a prestudy. He is often also the requirements administrator. The first thing that happens is that the idea (requirement) shall be the foundation of a number of requirements on the product. The requirements might be found through an arranged elicitation. For the elicitation all stakeholders are gathered and all requirements found are submitted to the database. The requirements administrator of the prestudy sees to that every requirement gets estimated according to the process proposal.

When all requirements have been estimated they are updated in the database and moved into the state investigated. With database tool support a high-level requirement specification is produced which shall contain all requirements and their

estimates. The high-level requirements specification is then presented to the product board that decides whether the prestudy shall lead to a project or not.

Project planning

A project manager is made responsible for the project, which is now in its planning phase. If needed, more requirements are elicited and estimated and some of the already existing requirements are split into more detailed requirements.

When all requirements have been estimated they are assigned the state specified and prioritized. It may be impossible to implement all of them, within the available time and with available resources, and therefore it is good to know which ones are more important than others. In this phase the project manager and the project steering group is responsible for both elicitation and prioritization. All specified requirements are gathered and a requirements specification is produced. The requirements specification is validated and then used for project planning and design.

When a requirement eventually has been implemented and validated by the customer it gets the end state implemented.

If a requirement somewhere along the process gets rejected it gets the end state rejected and is no longer considered.

Figure 5.6 below describes scenario 1. Phases, activities, states and outputs are further illustrated.

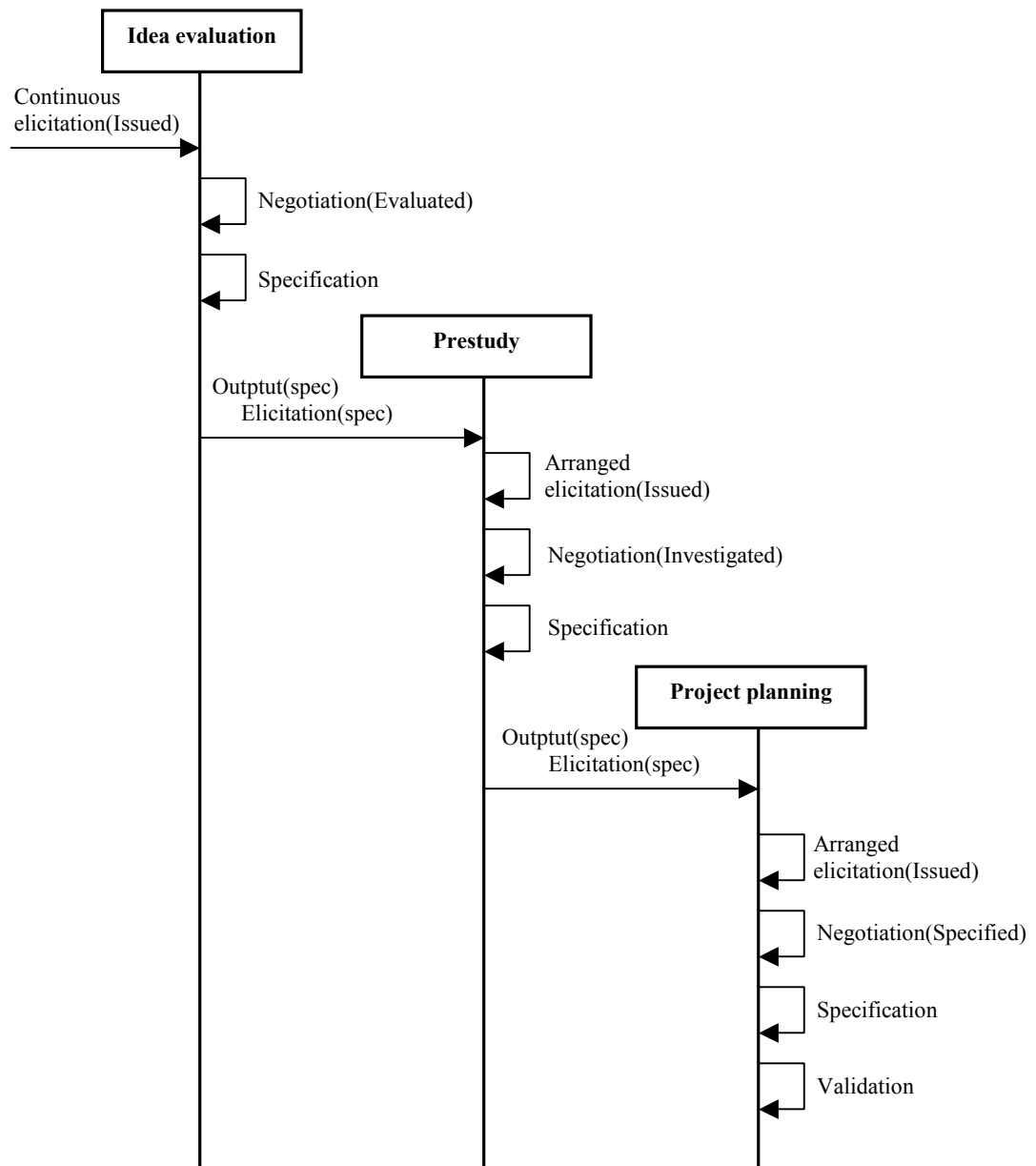


Figure 5.6 Phases, activities, states and outputs for scenario 1

2. Idea about a new feature for a product already under development

When an idea, about a special feature or new function for a product that is already in its project planning phase, appears, the way of handling it differs a bit from the way to handle totally new product ideas.

Depending on where the idea comes from there are two possible ways for it to join the project. If the idea comes from outside the project the idea is submitted to the database and it gets evaluated like all ideas, it gets the state evaluated. The idea evaluation report is presented to the project manager and if needed the project steering group that will approve or reject the idea. If it gets approved the project manager and the project steering group makes the required estimations, refines the requirement and updates the database and move the requirement straight into the specified state. This means that no prestudy will be performed. When it is in the specified state it will be handled like all requirements in the project planning phase.

If the idea comes from the project steering group itself or someone close to them, it is possible that they submit a requirement that must be part of the system. They make all estimations necessary and submit the requirement into the database directly as specified.

Figure 5.7 below describes scenario 2, case one, when someone outside the project submits the idea. Phases, activities, states and outputs are further illustrated.

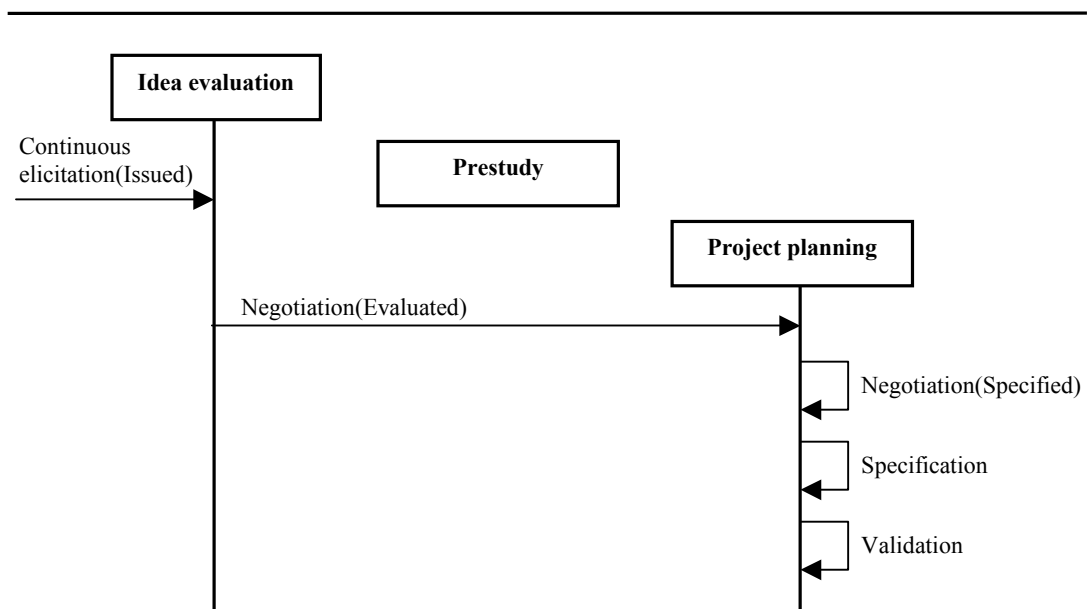


Figure 5.7 Phases, activities, states and outputs for scenario 2, case one

Figure 5.8 below describes scenario 2, case two, when someone inside the project submits the requirement. Phases, activities, states and outputs are further illustrated.

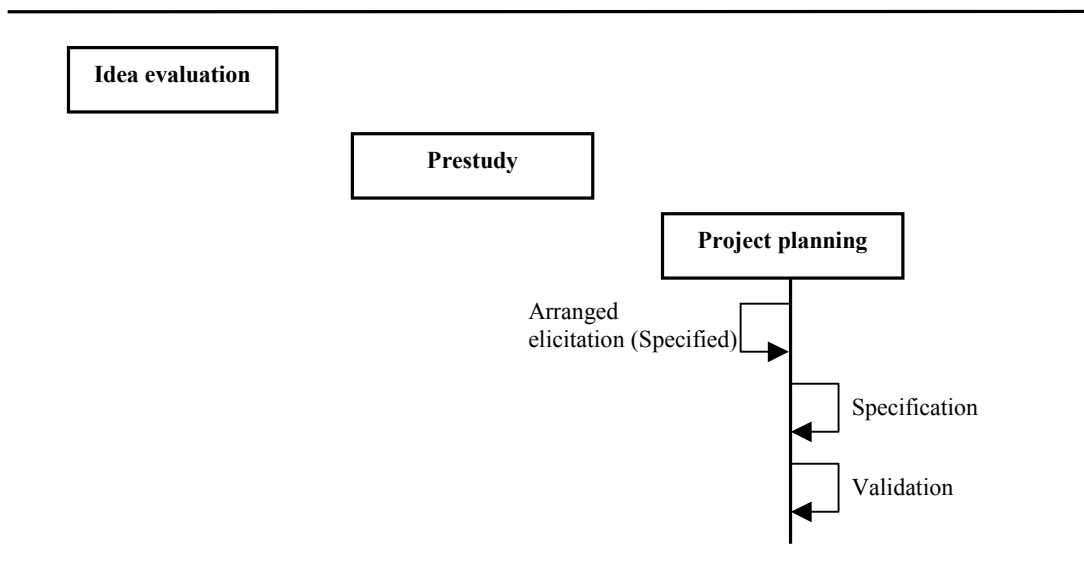


Figure 5.8 Phases, activities, states and outputs for scenario 2, case two

3. Ideas for a new release of an existing product

Ideas are continuously gathered through the web tool and stored in the database. Several ideas are about new features for an existing product that might be possible to implement in a new release of the product.

The requirements administrator gathers all issued ideas and sees to that they get estimated and updated to the evaluated state. He/she then presents the result to the product board that decides if a prestudy is needed or if the project shall start directly. They also have the possibility of rejecting the ideas. If they decide on a project all the selected ideas are treated as requirements for the new release and new requirements get elicited through an arranged elicitation. The requirements get estimated and refined and put into the state specified.

If they are unsure about some things they may decide to run a prestudy. Additional requirements get elicited and estimated and submitted to the database. A high-level requirement specification is produced and presented to the product board. They then again have to consider if a project shall start or not.

Figure 5.9 below describes scenario 3 case one, new release decision without prestudy. Phases, activities, states and outputs are further illustrated.

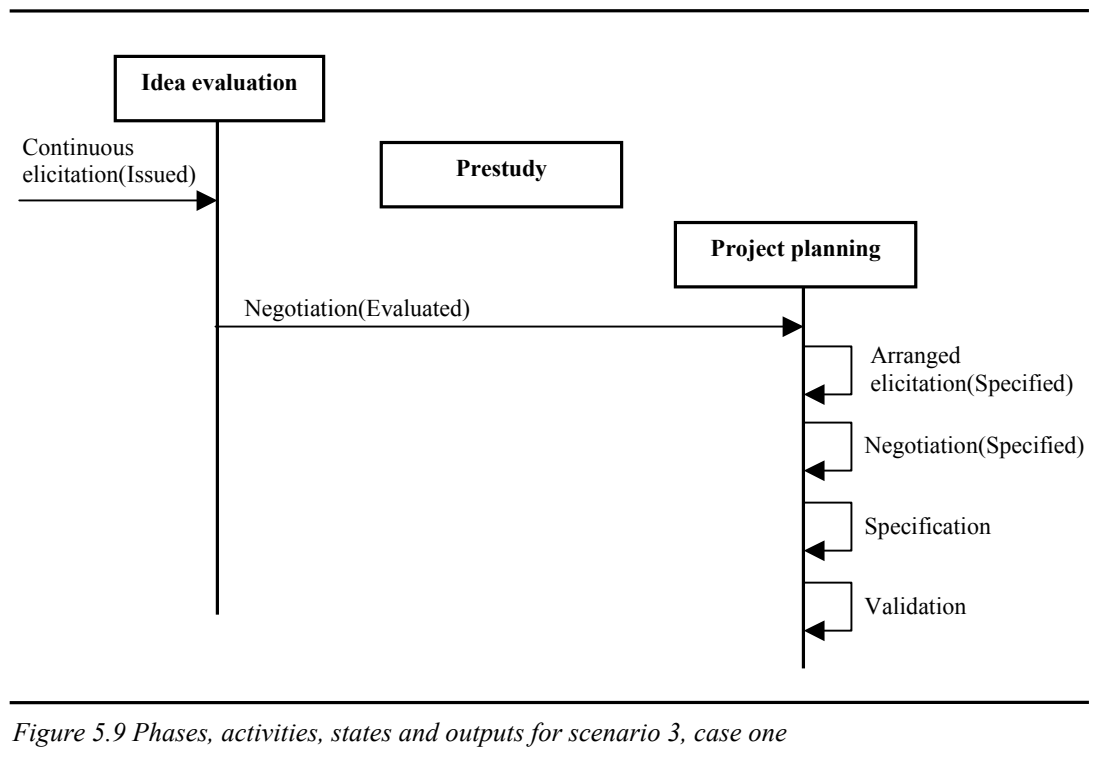


Figure 5.10 below describes scenario 3 case two, new release decision with prestudy. Phases, activities, states and outputs are further illustrated.

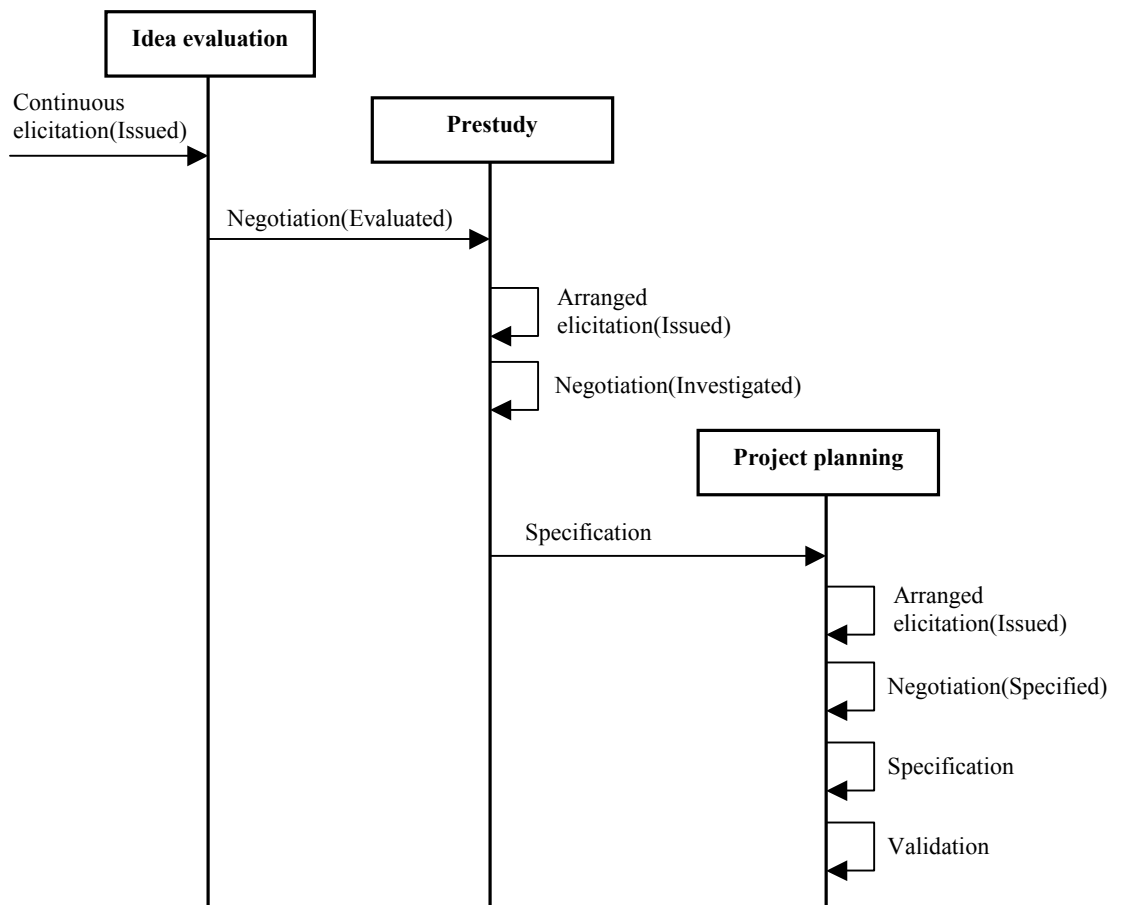


Figure 5.10 Phases, activities, states and outputs for scenario 3, case two

The last two chapters have described our process proposal, a database prototype and scenarios that supports and further explain the proposal. The next chapter describes how to evaluate the proposal if deployed.

6 Process Proposal Evaluation Plan

When introducing new processes to companies it is good to somehow evaluate if it has lead to the expected improvement or not [Sommerville, 2001]. This chapter describes how we have created a process proposal evaluation plan for the process we have proposed to C Technologies in chapter 4, and how it shall be used if the process is introduced.

6.1 Background

Having provided C Technologies with the improved process proposal, C Technologies must decide whether to deploy it or not. If they decide to deploy our process proposal into the organization it is important that they somehow can evaluate the process and see if it has led to the expected goals stated in section 4.1. Therefore a process proposal evaluation plan has been produced which is presented in the following sections.

Being able to evaluate the process after it has been deployed requires some kind of data collection and analysis. There are several different ways to choose what data to gather and how to use it [Humphrey, 1990]. The following are very important things to consider:

- Why do we need to measure our process?
- What questions help us reach the goals?
- There must be a clear objective for every measure
- Measures are useful in a long-term perspective
- Measures must have senior management support

Metrics and measurement can be used to understand development, control projects or improve processes [Fenton, Pfleeger, 1997]. Since our thesis regards process improvement we have concentrated on the latter.

A useful method that is used to find metrics is the Goal-Question-Metric method developed by Basili [Fenton, Pfleeger, 1997].

6.2 The Goal-Question-Metric method

The Goal-Question-Metric (GQM) method is suitable to use when evaluating if something has turned out as expected. The purpose is to find and present relevant metrics that C Technologies can use to evaluate the proposed process.

We have in chapter 4.1 defined goals with the process and with this evaluation plan C Technologies shall be able to determine whether these goals have been met or not.

The GQM method defines three steps to test if goals and objectives have been fulfilled [Fenton, Pfleeger, 1997]:

1. Define goals and objectives, what do we need to learn or know?
2. Generate questions that provide answers about whether the goals have been met or not.
3. Analyze the questions to find what metrics you need to answer each question.

6.2.1 Goals

The goals from chapter 4, is the ones we want to evaluate. The goals are:

1. To make all projects follow the same process
2. To make everyone aware of theirs and others responsibilities
3. To make new employees join projects easier and faster
4. To make the requirements better defined
5. To make the product definitions more detailed
6. To see to that all tasks get carried out
7. To make the elicitation more effective
8. To make sure that more ideas are gathered and documented
9. To make the prioritization more effective
10. To facilitate the finding of the most important requirements
11. To involve all stakeholders
12. To make sure that all ideas are evaluated

6.2.2 Questions

To find metrics that can give answers the following questions have been defined. In order to get full information there might be more than one question to each goal:

Question for goal 1 (To make all projects follow the same process)

Are all projects following the process?

Question for goal 2 (To make everyone aware of theirs and others responsibilities)

Do the employees know their responsibilities?

Do the employees know others responsibilities?

Questions for goal 3 (To make new employees join projects easier and faster)

Are new project members faster introduced?

Questions for goal 4 (To make the requirements better defined)

Are the requirements well defined?

Are the requirements easy to understand?

Has the requirement all its attributes?

Questions for goal 5 (To make the product definitions more detailed)

How well are products defined?

How many requirements does the requirements specification include?

Is the requirement easy to understand?

Has the requirement gotten all its attributes when it is presented in the requirements specification?

Question for goal 6 (To see to that all tasks get carried out)

How many activities get carried out?

Questions for goal 7 (To make the elicitation more effective)

How many requirements are found through continuous elicitation?

How many requirements are found per arranged elicitation?

Are all stakeholders present at arranged elicitation?

Are the stakeholders experienced in elicitation?

How much time is spent on elicitation per project?

Question for goal 8 (To make sure that more ideas are gathered and documented)

How many ideas are gathered?

Questions for goal 9 (To make the prioritization more effective)

How many requirements are given top priority?

How many of the top priority requirements are implemented?

How much time is spent on prioritization per project?

Are all stakeholders represented at prioritization?

How many requirements with low priority get implemented?

Questions for goal 10 (To facilitate the finding of the most important requirements)

How many requirements are given top priority?

How many of the top priority requirements are implemented?

How many requirements with low priority get implemented?

Question for goal 11 (To involve all stakeholders)

Are all defined stakeholders involved in the process?

Questions for goal 12 (To make sure that all ideas are evaluated)

Do all ideas go through the idea evaluation phase?

How many ideas get registered?

How many ideas get rejected?

How many ideas get approved?

6.2.3 Metrics

To get answers to the questions above the following list of metrics have been produced. For every question there is a number of metrics that together bring answers.

For some metrics it is useful to make the measure on a scale. We recommend that the scale is made with three values. The metrics that somehow measure activities should be collected in each phase.

Metrics for goal 1 (To make all projects follow the same process)

Number of projects started.

Number of projects using the process.

Metrics for goal 2 (To make everyone aware of theirs and others responsibilities)

Measure on a scale to what level an employee is aware of his/her responsibilities.

Measure on a scale to what level an employee is aware of other project members' responsibilities.

Metrics for goal 3 (To make new employees join projects easier and faster)

The time for a new employee between the employment date and the date he/she has become an independent project participant.

Metrics for goal 4 (To make the requirements better defined)

Measure if the requirement has all its attributes for each phase.

Measure on a scale the understandability of the requirement.

Number of used attributes for each requirement.

Number of possible attributes.

Metrics for goal 5 (To make the product definitions more detailed)

Number of requirements per requirements specification.

Measure on a scale the level of detail.

Measure on a scale the understandability of the requirement.

Number of used attributes for each requirement.

Number of possible attributes.

Metrics for goal 6 (To see to that all tasks get carried out)

Number of activities performed per project.

Number of activities that should be performed.

Metrics for goal 7 (To make the elicitation more effective)

Number of requirements found in elicitation per project.

Number of brainstorming occasions.

Number of requirements found during brainstorming

Number of focus group occasions.

Number of requirements found during focus group.

Number of new requirements in the database per time unit.

Number of participants in arranged elicitation.

Number of elicitation participants has been present at.
Time spent per arranged elicitation.
Number of requirements submitted to database after elicitation.

Metrics for goal 8 (To make sure that more ideas are gathered and documented)

Number of ideas collected in the database per time unit.

Metrics for goal 9 (To make the prioritization more effective)

Number of requirements with top priority.
Number of implemented requirements with top priority.
Time spent on prioritization per project.
Number of stakeholders present per prioritization.
Number of implemented requirements with low priority.
Number of prioritizations where pair-wise comparison is used.

Metrics for goal 10 (To facilitate the finding of the most important requirements)

Number of requirements with top priority.
Number of implemented requirements with top priority.
Number of implemented requirements with low priority.

Metrics for goal 11 (To involve all stakeholders)

Number of involved stakeholders.
Number of stakeholders that should be involved.

Metrics for goal 12 (To make sure that all ideas are evaluated)

Number of ideas that get evaluated.
Number of new ideas registered per time unit.
Number of ideas that get rejected.
Number of ideas that get approved.

Working with the GQM method requires both effort and time and 12 goals generates too many metrics to evaluate. Therefore the goals have been prioritized and only a few metrics of the ones proposed have been chosen for the first evaluation. These are presented in section 6.3.

6.3 Metrics recommendation

The twelve original goals generate too many metrics as mentioned above. Even if all metrics presented above are important it is impossible and unrealistic to collect them all in each project. Therefore we have prioritized the goals and selected the most relevant metrics.

The goals with the highest priority are the following:

1. To make the product definitions more detailed
2. To facilitate the finding of the most important requirements

3. To involve all stakeholders
4. To make sure that all ideas are evaluated

The goals have been chosen since we believe they are considered most relevant to the current situation at C Technologies. We believe that products need to be better defined by finding more requirements and prioritizing them after importance. We also believe that involving more stakeholders will increase communication between project members. The last thing is that it is also important to make sure that all ideas that are documented get evaluated. The metrics shall be used as indicators. They shall help provide information both about how the process has been working so far and how it will work in the future.

According to the questions presented in section 6.2.2 and metrics in section 6.2.3 the following metrics are considered to be the most important to C Technologies when evaluating the process:

1. Number of requirements per requirements specification.
Number of used attributes for each requirement.

How to collect the metrics and motivation: For every project the number of requirements in the final requirements specification shall be counted. This is done since we believe that the larger number of requirements there is, the better is the product specified. As the process is deployed to the organization and is used for every project it might be possible to find patterns that shows that projects with well-defined requirements are easier accomplished. The reason to count the number of attributes is that requirements containing all their attributes are better specified than requirements where attributes are missing.

2. Number of requirements with top priority.
Number of implemented requirements with top priority.

How to collect the metrics and motivation: The number of top prioritized requirements in every requirements specification shall be counted. This can at a later stage be compared to the number of top-prioritized requirements that really has been implemented. From this it will be possible to find patterns for how many requirements it is possible to implement in every project. It can also be of help when planning projects in the future and to know how many requirements that should be given top-priority.

3. Number of involved stakeholders.
Number of stakeholders that should be involved.

How to collect the metrics and motivation: Involving the stakeholders as proposed in the process should lead to increased communication between the project members. Counting the number involved stakeholders per project and then comparing the numbers against two things:

- The difference between the suggested number stakeholders and the actual.
- The outcome of the project in mind can help evaluate whether it is preferable to have good communication or not.

4. Number of new ideas registered.
Number of ideas that get evaluated.

How to collect the metrics and motivation: All ideas that are submitted to the database shall be counted regularly. This means that the number of requirements with the state issued shall be compared to the number of requirements with other states. It is important that the number of issued requirements do not grow to a number significantly higher than the number of requirements with other states. This is a way to predict if the requirements engineering process works and if it is good to use a database to store the requirements.

The evaluation plan should be used on a regular basis to collect the data continuously. In the future it might be possible to automate the collection of the metrics.

The fact that not all projects are of the same size must be considered when analyzing the collected data.

It is appropriate to combine the GQM method with a questionnaire where opinions about the process are gathered. The users of the process must also provide relevant feedback about how they experience the process so that adjustments can be made to satisfy their needs.

7 Summary and Further Work

This chapter summarizes our thesis and presents further work that may be realized. During our work we have gained an understanding about C Technologies methods and procedures, which has been very interesting. C Technologies is still a very young company with high potential, and with a better capability to structure their work and define their products they will most likely reach better efficiency and great success.

Our goal was to provide C Technologies with a process proposal regarding Requirements Engineering and this has been accomplished.

7.1 Summary

The result from the current situation analysis was that there is a need of a more structured requirements engineering within C Technologies. We used the conclusions and results from the current situation analysis to create a requirement specification for the process.

7.1.1 Process Improvement Proposal

Using the requirements specification a requirements engineering process was produced. The requirements engineering process covers the following phases:

- Idea evaluation
- Prestudy
- Project planning

Each phase includes the following activities:

- Elicitation
- Documentation
- Estimation
- Prioritization
- Specification
- Validation

To summarize the findings from our study of process models we list what we believe most necessary in a requirements process model.

- Phases
- Activities
- How to store and document requirements
- The attributes a requirement shall contain
- Requirements states
- How requirements are collected
- Actors participation and responsibilities

- Stakeholders participation and responsibilities
- Estimations that shall be conducted for the requirements

7.1.2 Process Proposal Evaluation Plan

It has not been part of our work to deploy the process to the organization. Therefore an evaluation plan for the process proposal has been created. Using the GQM method C Technologies shall be able to evaluate the process if deployed.

The plan contains the goals for the process, questions that have to be answered to verify that the goals have been met, and finally metrics that will provide answers to the questions.

In the process improvement proposal chapter 12 goals with the process was presented. When producing the process proposal evaluation plan we found that these goals were too many. Therefore the following 4 goals were chosen for evaluation of the proposed process. The goals are:

1. To make the product definitions more detailed
2. To facilitate the finding of the most important requirements
3. To involve all stakeholders
4. To make sure that all ideas are evaluated

To be able to evaluate these goals the following metrics were chosen:

1. The number of requirements per requirements specification shall be counted.
2. The number of used attributes for each requirement shall be counted.
3. The number of requirements with top priority shall be counted.
4. The number of implemented requirements with top priority shall be counted.
5. The number of involved stakeholders shall be counted.
6. The number of stakeholders that should be involved shall be counted.
7. The number of new ideas registered shall be counted.
8. The number of ideas that get evaluated shall be counted.

The metrics are to be used as indicators to evaluate both how the process is working now and how it will work in the future.

7.2 Further work

Due to the time and resource limits we have been forced to leave some work out:

First of all the process proposal must be approved for deployment. Since the proposal is the very first introduction of a complete requirements engineering process it is probable that some adjustments must be made. This also includes making decisions about how the process shall be deployed and time and resources available.

We believe that educating the employees at C Technologies, who are affected by the process, is the first step towards deployment. How to educate the personnel is regarded as further work.

How the work in the different activities shall be documented is up to C Technologies to decide. This includes how meeting protocols shall be written and frameworks for metrics collection.

The database prototype we have developed includes the basis for what we think a database shall contain. It is recommended to investigate if there exists a CASE tool that suits our process proposal, and alternately investigate the possibility to develop a database with all required functions. Cost should be estimated in both cases.

The process proposal does not specify how the requirements shall be presented. Further work is to create frameworks for requirements specifications.

The CASE tool Focal Point should be further investigated and evaluated by affected stakeholders.

Change Management within requirements engineering should be investigated and added to the process.

8 Glossary

A

Active Server Pages

Makes it possible to create interactive web pages

ASP

See Active Server Pages

B

Baldrige award

An award given to companies that achieves a certain level of quality

Brainstorming

Meeting with the goal to extract ideas/requirements regarding a predetermined issue

C

Capability Maturity Model

A model or framework for determining and increasing the maturity level of a company

CASE Tool

Computer Aided Software Engineering tool. Automatic tools, which can be of help in different software development areas

Closed question

A question to which the selection of answers is already defined by the one who ask the question

CMM

See Capability Maturity Model

Commercial Of The Shelf products

Soft- and hardware products that can be bought in a store

Context diagram

A way to visualize requirements

Cost-value based prioritization

Prioritization where cost and value are considered

COTS

See Commercial Off The Shelf products

D

Data-flow diagram

A way to visualize requirements

Data model	A way to visualize requirements
E	
Elicitation	Basic activity in the requirements engineering process where requirements are obtained and collected
End-user	The final recipient of the software product
Entity/Relationship model	A way to design a database
F	
Feasibility Study	An investigation to determine if something is possible
Feature style	A way to present a requirement
Focal Point	A CASE tool that handles requirements among other things.
Focus group	A method to elicit requirements
Framework	A predetermined way to present something. A help for the user.
G	
Goal-means analysis	Technique for checking if all requirements have been found
Guidelines	Instruction of how to perform something
H	
High-level requirement	Requirement with low level of detail
I	
Information matrix	Way to present data from a survey or investigation
ISO	The International Organization for Standardization
Issuer	A person that comes up with a

	requirement or idea
K	
Key Process Area	Area on where to put effort, according to the CMM
M	
Maintainability	Software quality issue stating that software must be possible to evolve to meet the changing needs of customers
O	
Open-ended question	A question to which an answer in own words can be given
P	
Pair-wise comparison	To compare two requirements against each other, regarding a specific criteria
Prioritization	A basic activity in requirements engineering where requirements are given a certain priority
Process model	A model of how software is developed in an organization
Product Board	Group of employees at C Technologies, responsible for decision making regarding new products
Prototyping	A software development process where versions of the software is successively built to help elicit requirements
Q	
Qualitative	Something that varies in kind
Quality Assurance	Method to make sure a certain level of quality has been reached
Quantitative	Something that varies in amount
Questionnaire	A list of questions to be answered by a number of people as a part of a survey

R	
RDEM	Requirement Driven Evolution Model, developed by Ericsson
RE	See Requirements Engineering
REPEAT	Requirements Engineering Process At Telelogic, developed by Telelogic
Requirements Engineering	The part of a development process where requirements are handled
Requirements Management Group	Group responsible for handling requirements at C Technologies
Respondent	Person answering the questions in a questionnaire
RQMG	See Requirements Management Group
S	
Scenario	An imagined sequence of future events
Software Engineering	Engineering technique comprising theories, methods, techniques, and tools to develop large-scale complex software systems
Spiral Model	A software or requirements process model where several development techniques are incorporated into one process
SQL	Standard Query Language, language which makes it possible to communicate with a database
Stakeholder	A person with interest in a certain issue
Survey	An investigation of behavior and opinions
T	
Throwaway prototyping	A software development process model where versions are developed to elicit requirements. The prototype is discarded

before actual development commences

U

UML

Unified Modeling Language. Formal language used to specify, visualize, and document the artifacts of an object-oriented system under development

Usability

Software quality issue stating that software must be easy to use and support the user

Use case

Way to visualize requirements

V

Validation

Basic activity in the requirements engineering process where requirements are checked for consistency and completeness before development begins

9 References

- [Carlshamre, Regnell, 2000] Carlshamre, P., Regnell, B., (2000), *Requirements Lifecycle Management and Release Planning in Market-Driven Requirements Engineering Processes*, accepted for publication at REP2000 (International Workshop on the Requirements Engineering Process: Innovative Techniques, Models, and Tools to support the RE Process, 6th-8th of September 2000, Greenwich UK, proceeding the DEXA Conference) © IEEE CS Press.
Comments: Comparison of two requirements engineering processes, RDEM at Ericsson and REPEAT at Telelogic.
- [Christensen, 1997] Christensen, L. B., (1997), *Experimental methodology*
Comments: Describes different research methods.
- [Connolly et al, 1999] Connolly, T., Begg, C., Strachan, A., (1999), *DATABASE SYSTEMS, A Practical Approach to Design, Implementation and Management*, Addison-Wesley
Comments: Practical book about how to design relational databases.
- [Davis, 1993] Davis, A. M., (1993), *Software requirements – objects, functions and states*, Prentice Hall
Comments: Focuses on the early phases of software development lifecycle commonly called software requirements analysis or software requirements specification.
- [Feigenbaum, 1994] Feigenbaum, A.V., (1994) “*Total Quality Management*”, Encyclopedia of Software Engineering, edited by J.J. Marciniak, John Wiley & Sons, 1994, pp. 1359-1360.
Comments: 10 definitions of quality.
- [Fenton, Pleegeer, 1997] Fenton, N. E., Pleegeer, S. L., (1997) *Software metrics – A rigorous & practical Approach*, 2nd Edition.
Comments: Book that covers software metrics. Includes why, how, when and who.
- [Focal Point, 2001] www.focalpoint.se
- [Holme, Solvang, 1997] Holme, I. M., Solvang, B. K., 2nd Edition (1997), *Forskningsmetodik – Om kvalitativa och kvantitativa metoder*, Studentlitteratur.
Comments: Explains fundamental concepts and terms regarding qualitative and quantitative research methods. Important things to think about when planning, performing and analyzing studies. Few

examples and a comprehensive explanation make the book too much all embracing. It contains more examples and more thorough explanations regarding quantitative methods than for qualitative methods

[Humphrey, 1989]

Humphrey, W. S., (1989) *Managing the Software Process*, Addison Wesley.

Comments: Practical guide for improving the software development and maintenance process.

[Höst et al, 2000]

Höst, M., Regnell, B., Natt och Dag, J., Nedstam, J., Nyberg, C., (2000), *Exploring Bottlenecks in Market-Driven Requirements Management Processes with Discrete Event Simulation*, Published in Proceedings of PROSIM2000 (Software Process Simulation Modeling Workshop, London, July, 2000) © Prosim.

Comments: Presents a simulation of a market-driven requirements management process.

[Karlsson, 1996]

Karlsson, J., (1996), *Software Requirements Prioritizing*, IEEE.

Comments: Why prioritize? The article considers how to calculate accuracy of the comparison. Comparisons from an industrial case study at Ericsson Radio Systems, where two different approaches on prioritization have been made. Numerical prioritizations and pair-wise comparisons have been compared. The latter is better in many ways.

[Kotonya, Sommerville, 1997] Kotonya, G., Sommerville, I., (1997), *Requirements engineering processes and techniques*, John Wiley & sons.

Comments: Covers almost everything about requirements. Easy to understand and contain lots of good tips and help for how to handle the requirement process and improvement efforts.

[Lauesen, 2000]

Lauesen, S., 2nd Edition (2000), *Software requirements – Styles and Techniques*.

Comments: Chapter 4 is relevant. It describes different elicitation techniques. As for the rest the book contains mostly requirement representation techniques.

[Natt och Dag et al, 2001]

Johan Natt och Dag, Björn Regnell, Pär Carlshamre, Michael Andersson, Joachim Karlsson (2001), *Evaluating Automated Support for Requirements Similarity Analysis in Market-Driven Development*.

Comments: The paper is about evaluating automatic comparison and textual analysis between requirements, in order to find duplicates.

- [Pagano, 1994] Pagano, R. R., 4th Edition (1994), *Understanding statistics in the behavioral sciences*, West publishing company
Comments: Explanations of statistical methods and calculations in plain text with examples.
- [Paulk et al, 1993] Paulk, M. C., Curtis, B., Chrissis, M. B., Weber, C. B., (1993), *Capability Maturity Model version 1.1*.
Comments: The book about CMM, complete description.
- [Regnell et al, 1998] Regnell, B., Beremark, P., Eklundh, O., (1998), *A Market-driven Requirements Engineering Process – Results from an Industrial Process Improvement Programme*, Published on Proceedings of CEIRE'98 (Conference on European Industrial Requirements Engineering, London, 19-20 October 1998).
Comments: Describes continuous requirements elicitation and prioritization together with expert cost estimation as a basis for release planning.
- [Robertson, Robertson, 1999] Robertson, S., Robertson, J., (1999), *Mastering the requirements Process*, Addison Wesley.
Comments: Describes software requirements, processes, requirement types etc.
- [Sommerville, 2001] Sommerville, I., (2001), *Software Engineering*, 6th Edition, Addison Wesley.
Comments: Presents a broad perspective on software systems engineering.
- [Steeple, 1994] Steeples, M. M., (1994), *The Baldrige Award and ISO 9000 in the Quality Management Process*, Published in IEEE Communications magazine, October 94 on page 52
Comments: Article about ISO and the Baldrige award.
- [Wiegers, 1999] Wiegers, K. E., (1999), *Software requirements*, Microsoft.
Comments: Describes software requirements, processes, requirement types etc.

Appendix A

CMM – Requirements Management

Appendix A further describes the CMM activities that specifically address Requirements Management.

Regarding requirements management CMM describes two major *goals*, one *commitment* to perform, four abilities to perform and three activities to perform [Paulk et al, 1993].

The goals are:

1. *System requirements allocated to software are controlled to establish a baseline for software engineering and management use.*
2. *Software plans, products and activities are kept consistent with the system requirements allocated to software.*

The commitment is:

1. *The project follows a written organizational policy for managing the system requirements allocated to software.*

The abilities are:

1. *For each project, responsibility is established for analyzing the system requirements and allocating them to hardware, software, and other system components.*
2. *The allocated requirements are documented.*
3. *Adequate resources and funding are provided for managing the allocated requirements.*
4. *Members of the software engineering group and other software-related groups are trained to perform their requirement management activities.*

The activities are:

1. *The software engineering group reviews the allocated requirements before they are incorporated into the software project.*
2. *The software engineering group uses the allocated requirements as the basis for the software plans, work products and activities.*
3. *Changes to the allocated requirements are reviewed and incorporated into the software project.*

This is the concept of CMM, it specifies what to do but not how to do it.

Appendix B

Questionnaire

Appendix B presents our questionnaire used for the current situation analysis.

The questionnaire presented below is the one we handed out to the selected respondents. Here it also contains the answers we received. In front of every answering alternative we present the respondent frequency for that alternative. Open questions are presented without any changes or refinements.

20 questionnaires were handed out and 14 of the selected respondents answered.

The questionnaire from the Patent department is unfortunately not useful since the patent department is not working with the kind of projects that this investigation regards. Due to this the answers from the patent department are not considered in the analysis of the answers. This means that maximum respondent frequency of the questions is 13.

Questionnaire

Requirements engineering at C Technologies

High-level requirement process

Urban Martinsson and Åsa Karlsson

Introduction

This questionnaire is part of our, Åsa Karlsson's and Urban Martinsson's, Bachelor Thesis. The main goal of our Thesis is to provide C Technologies with a process proposal for high-level requirement activities with emphasis on elicitation, prioritization and decision-making. In order for us to be able to produce a proposal we need to investigate how projects are accomplished today, which activities that work well and which needs improvement.

If you are working with high-level requirements, or are somehow affected by them, we believe that you are the right person to answer our questions. The aim of our work is to make a proposal that will be of use to you and this is why we need your help. Our expectation is that the proposal will help structure, define and facilitate activities relevant for your high-level requirement process.

You are welcome to answer the questions in Swedish or English. We estimate that it will take you approximately 30 minutes to complete the questionnaire.

Terminology

- **Elicitation** - Elicitation is the part of a requirement process where the requirements are found and expressed. Analyzing ideas and thoughts about the product to be developed and making requirements out of the wishes. The customer's ideas and thoughts are of major importance and the persons documenting the elicited requirements must seriously concern these. The elicitation can be performed in several different ways, examples are interviews, brainstorming and prototyping.
- **Process / Work models** – Defines a way to work. For example a certain order or a checklist to follow when executing one or more activities. The purpose of this is to be able to perform the activities the same way over again, within different projects, using guidelines for how to do it.
- **Prioritization** – Often it is not possible to implement all the requirements elicited, in this case the requirements have to be prioritized. Decisions have to be made about which of the requirements that shall be implemented. Maybe the requirements with low prioritization have to wait for the next release, or maybe they never will be implemented at all.
- **High-level requirements** – High-level requirements are the requirements elicited on an early stage in a project. They can be ideas about what functions might be useful. The high-level requirements will be refined later on in the process.
- **COTS** – Commercial Of-The-Shelf products. Products to be sold to a broad category of customers.

Respondent information

My position

- | | | |
|--|--|---|
| <input type="checkbox"/> 3 Project manager | <input type="checkbox"/> 1 Sales | <input type="checkbox"/> 2 Marketing |
| <input type="checkbox"/> 4 Development | <input type="checkbox"/> 1 Patent department | <input type="checkbox"/> 0 Support |
| <input type="checkbox"/> 1 Management | <input type="checkbox"/> 1 Purchasing department | <input type="checkbox"/> 1 Other = <i>Quality</i> |

I have been working at C Technologies for

- | | | |
|---------------------------------------|--|--|
| <input type="checkbox"/> 1 0-2 months | <input type="checkbox"/> 1 2-6 months | <input type="checkbox"/> 5 6-12 months |
| <input type="checkbox"/> 1 1-2 years | <input type="checkbox"/> 6 More than 2 years | |

Processes and activities

1. Are projects planned according to specific work models or development processes?

☐ 2 Yes ☐ 8 Partly ☐ 1 No ☐ 2 Do not know

1.1. If yes or partly, what kind of work models or development processes is used?
(Waterfall model, spiral model etc. Please describe it)

Answers: Comments:

Yes	<i>Utveckla egen WAC, blandning av Ericssons PROP och Telelogics metod, mål att den skall användas av hela koncernen, skalbar process, skall vara lätt att sätta sig in i för nya anställda</i>
Partly	<i>C Technologies own model</i>
Partly	<i>TTM o TTC (Time to market and Time to customer) models using Pre-, Feasibility- and R&D phases with milestones</i>
Partly	<i>Internal close to Ericsson's PROPS</i>
Partly	<i>Vattenfall, C Technologies current version</i>
Partly	<i>En process som finns beskriven i dokument. Förstudie-utveckling-verifiering-validering.</i>
Partly	<i>Vattenfallsmodell</i>
No	<i>Our development process is not used.</i>
Do not know	<i>Vi har en egen projektmodell, men jag vet inte om den baseras på någon specifik känd modell</i>
Partly	<i>This is currently being implemented (I don't know what the outcome will be)</i>

2. How many simultaneous development projects are normally in progress at C Technologies, pre-studies excluded?

☐ 6 0-5 ☐ 5 5-10 ☐ 1 10-15 ☐ 0 15-20 ☐ 0 More than 20

3. What different kinds of projects are performed at C Technologies?

☐11 Customer specific ☐10 Technical projects (In-house) ☐9 COTS development
☐7 Re-Spin development ☐1 Other = *Improvement project within the company*

4. Is a process followed when working with different high-level requirement activities, such as elicitation, prioritization, specification writing etc?

☐1 Yes ☐8 Partly ☐2 No ☐2 Do not know

4.1. If yes, please describe the process and the activities included in the process.

Comments:

Vi måste förstå att vi definierar en marknad som inte finns. Mycket uppfinns av ingenjörer och sedan testas idén av marknaden

Vi har ingen modell men marknadsundersökningar och egna krav diskuteras

Specification och handhavande spec har börjat skrivas, (oftast), krav från marknadsavdelningen följs mindre ofta än krav från utv

4.2. If partly, which activities are performed?

☐5 Elicitation ☐6 Negotiation with customer ☐6 Prioritization
☐2 Documenting ☐1 Other = *Brainstorming and prototyping*

4.3. If no, how much would a defined requirement process facilitate your work?

☐0 Not at all ☐0 Little ☐1 Average ☐3 Much ☐0 Do not know

Please motivate

Answer: Comments:

Much Simplifies finalization of projects

Average Så länge alla känner sig tillräckligt delaktiga i processen och tillåts komma med input är det säkert mer till fördel än nackdel

Much *Bör vara skalbar, rätt anpassad, exempel, checklista för att man ej skall missa någonting*

Much *Will make things easier*

5. If a process is followed, is that process somehow documented?

☐ 0 Yes ☐ 2 Partly ☐ 7 No ☐ 1 Do not know

6. Who has the overall responsibility for the requirements within a specific project?

☐ 0 Management representative ☐ 8 Project Manager ☐ 2 Marketing representative
☐ 1 Other = *Not defined*

7. In a project, is someone working only with requirements?

☐ 1 Yes ☐ 9 No ☐ 3 Do not know

8. Are high-level requirements used for any kind of project planning?
(Cost, time, resources etc)

☐ 9 Yes ☐ 1 No ☐ 3 Do not know

8.1. If yes, for which parts of project planning are they used?

☐ 9 Resource estimation ☐ 9 Time estimation ☐ 8 Cost estimation
☐ 6 Technical possibilities ☐ 4 Competence needs ☐ 0 Other

8.2. If no, would it be useful to use the high-level requirements for project estimation?

☐ 2 Yes ☐ 0 Partly ☐ 0 No ☐ 0 Do not know

Please motivate

Answer: **Comments:**

Yes *En klar och tydlig kravspecifikation med bra prioriteringar är bra underlag för tidsuppskattning.*

Yes *If this will give a faster estimation of the project time and cost it would be useful.*

9. Do C Technologies have documented quality goals?

☐ 1 Yes ☐ 7 No ☐ 5 Do not know

9.1. If yes, do you know them by heart?

☐ 0 Yes ☐ 1 No

10. Is there a reason for working towards quality goals at C Technologies?

☐ 0 Not at all ☐ 0 Little ☐ 1 Average ☐ 12 Much ☐ 0 Do not know

Please motivate (What would it mean to you?)

Answer: Comments:

Much Growing organization with growing market demands in all areas

Average Om man diskuterar allt baserat på KUNDKVALITET, så ok

Much Kvalitet innebär produkthållbarhet och kunduppskattning

Much Less quick fixes close to or after project ends

Much Undvika omarbete, dålig service till kunder

Much De måste vara relevanta och konkreta, målen och hur man kommer dit skall specificeras

Much Higher customer satisfaction, better recourse management etc

Much Goals are mostly necessary to achieve things

Much We need to provide products with the right quality to satisfy customer needs

Much Kvalitet är mycket viktigt och hjälper till att definiera när projekten är slutförda.

Much Yes, the processes (if any) at C Tech are not well implemented. Most of the employees just "run like hell" to get things done.

11. Do specific projects have their own quality goals?

☐ 1 Yes ☐ 2 Partly ☐ 6 No ☐ 3 Do not know

Stakeholders and customers

12. Who are the stakeholders regarding high-level requirements?

- | | | |
|---|--|--|
| <input type="checkbox"/> 8 Management group | <input type="checkbox"/> 11 Marketing department | <input type="checkbox"/> 5 Customers |
| <input type="checkbox"/> 4 Support department | <input type="checkbox"/> 8 Development department | <input type="checkbox"/> 3 Purchasing department |
| <input type="checkbox"/> 2 Financial department | <input type="checkbox"/> 1 Other = <i>Everybody else</i> | |

13. Which categories of customers are C Technologies different products aiming at?

Comments:

Consumer

Everyone that are in contact with letters of any kind

Slutkund till 100 %, diskussioner om integrering där

Alla som har behov av att samla in informatioon, studenter, advokater, journalister, lärare, sjukvårdspersonal, you name it

Business professionals, students

Mest ungdomar studenter etc, på grund av prisbilden har tyvärr mest andra kunder blivit aktuella, C Tehc tror dock fortfarande att studenter är huvudfokus

Studenter, businessmen, alla som behöver översättning (15-60 år)

Endusers cpen (sttuodents, advokater, it people etc), Bank (oem), industri etc

Students, executives, OEM etc

End customers (consumer market), OEM

End-consumers bying C-Pen. OEM customers / business to business

Mobile peo, students, Early adopters

"Early adapters" just to be the fokus but now we aim for the mass market. Still some of our products aim for early adapters.

Elicitation

14. Is a specific process followed, when gathering or eliciting high-level requirements?

☐ 0 Yes ☐ 1 Partly ☐ 8 No ☐ 4 Do not know

14.1 If yes or partly, please describe the process used.

Answer: Comments:

Partly	Ideerna kommer från R&D via den som håller ihop idebanken, support ger sin input
Yes	Så att vi kan prioritera och göra rätt saker. Vi har inte möjlighet att göra allt utan måste satsa på det viktigaste.

14.2 If no, is there a need for a defined process for the high-level requirements elicitation?

☐ 9 Yes ☐ 0 Partly ☐ 0 No ☐ 1 Do not know

Please motivate

Answer: Comments:

Yes	<i>We need to refine so that we develop the "right" products</i>
Do not know	<i>Concept needs to be explained</i>
Yes	<i>Sena ändringar i kravspec dyker för ofta upp, ofta ej genomtänkta förslag, många möjligheter förbises</i>
Yes	<i>Makes life easier, minimize risks</i>
Yes	<i>To quickly come up with ROI, cost, time</i>
Yes	<i>When working according to a method (my experience is that) more work can be done in less time</i>

15. Are any particular methods such as brainstorming, observation, etc used for elicitation?

☐ 3 Yes ☐ 8 Partly ☐ 2 No ☐ 0 Do not know

15.1. If yes or partly, please describe the methods used.

Answer: Comments:

Yes Meetings

Partly Brainstorming

Partly Brainstorming är alltid bra, dock måste man ha en projekt ledare som brinner för att förverkliga ideerna (produkten)

Partly Brainstorming, Marknadsundersökningar, kommentarer från användare

Partly Brainstorming ibland

Yes Common brainstorming

Partly Partly reverse engineering

Partly Möten förekommer ibland för brainstorming. Att samla synpunkter och ideer inför projekten har varit syftet.

Yes Brainstorming, but only with C Tech people

15.2. If no, is there a need for defined methods when eliciting high-level requirements?

Not at all Little Average Much Do not know

Please motivate

Answer: Comments:

Average Works quite well now

Average We have more ideas than resources at the moment

Average Depends on context and intended use

Average The requirements are needed but there are different ways to achieve them

Much Strukturerat arbete mot att få kravspec på plats underlättar uppstarten av projektet. Finns det rutiner så snabbas förmodligen arbetet upp.

16. Does the eliciting activity differ in different projects?

☐ 4 Yes ☐ 0 Partly ☐ 5 No ☐ 3 Do not know

16.1 If yes or partly, please describe how they differ.

Answer: Comments:

Yes New people - new methods

Yes Mest tror jag det beror på vilken projekt ledare som ansvarar för projektet

Yes Varje projekt ledare gör på sitt sätt

Yes Since no method exist, the activity is based on the project managers personal experiences

17. Which persons perform the elicitation?

(Persons position)

Answers:

All involved

Mainly the management of respectively department and project people

Kreativa R&D personer

Various depending on projects

Projekt ledare, marknad, möjligen ibland någon utvecklare

Projekt ledaren tar hjälp av de han behöver

Project manager, technical stuff, market representative

Differ, most often it is feasibility study project leader.

Projekt deltagare

Respondent himself

Marketing and R&D

18. How many requirements are maximally elicited for a new product?

☐ 6 Less than 25 ☐ 1 25-50 ☐ 1 50-100 ☐ 0 100-200

☐ 0 More than 200 ☐ 5 Do not know

19. Are negotiations with the customer held when eliciting requirements?

Yes Partly No Do not know

19.1. If yes or partly, please describe how they are performed.

Answer: Comments:

Yes Hope so...

Partly Yes, if OEM partly through marketing and sales department, if C-Pen

Partly Discussions

Partly I OEM-projektet definieras ofta ett antal krav på funktioner och prestanda.

Prioritization

20. Are high-level requirements prioritized?

Yes Partly No Do not know

20.1. If yes or partly, please describe how they are prioritized.

Answer: Comments:

Partly Impossible req can get lower priority

Yes At least in operations

Yes Den bästa iden jag vet är att specen är manualen och / eller Datablad

Partly Not defined

Yes Genom diskussion, olika önkemål viktas mot varandra, stor hänsyn till tekniska möjligheter, så klart

Partly They set the framework for the project

Yes We have one page project contract form

Partly I de fall det förekommer krav prioriteras vissa bort pga att de anses inte ha tillräcklig nytta i förhållande till resursåtgång.

Yes What req. To be done/fulfilled first. Time order only

21. Does the prioritization activity differ in different projects?

Yes Partly No Do not know

21.1. If yes or partly, please describe the differences.

Answer: Comments:

Yes They need to since we develop different projects

Partly Chefer prioriterar alltid, det är deras jobb

Yes differences depending on project organization

22. Are any particular methods used for prioritization?

☐ 2 Yes ☐ 9 No ☐ 2 Do not know

22.1. If yes, please describe the methods used.

Comments:

Discussions and sometimes even fishbone or other weight methods

Perhaps no experience

22.2. If no, is there a need for defined methods when prioritizing high-level requirements?

☐ 1 Not at all ☐ 3 Little ☐ 2 Average ☐ 3 Much ☐ 1 Do not know

Please motivate

Answer: Comments:

Little Social, technical and entertaining done by those who will produce good

Average If its not taking away the "stomach feeling"

Not at all Man skall inte krångla till det mer, tre personer runt ett bord, mycket kaffe och en projekt ledare som håller ihop det brukar räcka

Average Perhaps, no experience

Do not know Probably but I have usually not been involved

Much Det behövs en process, krav och intressenter prioriteras efter olika kriterier

Little It is depending on the circumstances, customers, internal objectives/goals and resources. It will change from time to time.

Much Det underlättar säkert arbetet om det finns enkla och fungerande metoder för prioritering.

Not at all *We do not prioritize on this level*

Much *Again, I believe in structured working for better results*

23. Which ones of the following stakeholders take part in the prioritization?

- ☐ 11 Representative from Marketing department
- ☐ 4 Representative from Purchasing department
- ☐ 11 Representative from Management group
- ☐ 10 Representative from Development department
- ☐ 1 Representative from Financial department
- ☐ 1 Representative from Support department
- ☐ 4 Representative from Customer
- ☐ 2 Other = *Project manager*

24. Are requirements reformulated before prioritization?

☐ 0 Yes ☐ 2 Partly ☐ 6 No ☐ 5 Do not know

25.1. If yes or partly, please describe why the requirements are reformulated.

Answer: Comments:

Partly *When badly formulated*

Partly *Vid behov*

25. Approximately how much time is spent on prioritization of the high-level requirements?

Answers:

Too little

Enough

Lite men viktigt, detta kan vara chefens viktigaste jobb en dag var tredje månad då specen skall frysas

Lite nu och då i allmänna diskussioner

Mellan 5-100 timmar

Vet ej

Follow up does not exist

Differ from project to project

Ingen aning, väldigt lite < 1% av projekttid.

20 % of prestudy phase

26. Is too much time spent on prioritization?

☐ 0 Yes ☐ 0 Partly ☐ 10 No ☐ 3 Do not know

Please motivate

Answer: Comments:

No So far we deliver projects in time

No Snarare kanske för lite om man helar hela processen att utvärdera alternativ (krav)

No Det kan dock effektiviseras, sker väldigt luddigt idag

No More needed due to impact on project when prio is wrong.

No Good front up work will always gain time

No Borde satsa mer och tydligare på prioritering tidigt så att man gör de viktigaste nyttigaste delarna.

No If priorities are set at an early stage project may be finished faster (or earlier if nessesary). Most important req. Are fulfilled first.

27. What happens to the requirements that are rejected in the prioritization?

☐ 4 They never get implemented
☐ 5 The requirements are moved to the next release
☐ 1 Other = *sometimes implemented later, often never*
☐ 4 Do not know

28. Are requirements that are reformulated or changed late in the process reprioritized?

☐ 5 Yes ☐ 1 No ☐ 5 Do not know

Process measuring

29. Are the requirement activities somehow measured?

(Time, number of requirements handled, number of change requests etc)

Yes Partly No Do not know

30.1. If yes or partly, which measurements are performed?

30. Have you experienced any problems regarding the way high-level requirements are handled today?

Yes Partly No Do not know

31.1. If yes or partly, please describe the problems experienced.

Answer: Comments:

Yes Low visibility, not maintained, very low communication when changed, req not implemented by developer even if he/she say they are, req are changed by developer and no one knows about is

Partly Detta är det svåraste som finns då man skall definiera nya produkter, det SKALL vara svårt annars har man inte spämt bågen tillräckligt mycket

Partly Det saknas process samt dokumentationsunderlag

Yes Inte genomtänkta (-arbetade) därför tas de inte på allvar, ibland är syftet med vissa produkter oklart varför krav är svåra att prioritera / utvärdera

Yes Misunderstandings

Yes We miss some of the requirements due to poor projects (not performing continuos follow ups)

Yes Ingen som har ansvar för att organisera, visualisera och prioritera dessa.

Yes Requirements are change often, We have very little customer input. Most of the time it is management who have the desicion on what the requirements are.

Appendix C

Requirement specification

Appendix C presents our requirements specification produced to help us develop a process proposal tailored for C Technologies.

Requirement specification for the high-level requirement process at C Technologies.

1. Theoretical requirements

- 1.1. The process shall be adaptable.
- 1.2. The process shall be measurable.
- 1.3. The process shall be easy to understand.
- 1.4. The process shall be easy to use.
- 1.5. The process shall facilitate the producing of requirements that are:
 - 1.5.1. Traceable
 - 1.5.2. Correct
 - 1.5.3. Unambiguous
 - 1.5.4. Verifiable
 - 1.5.5. Consistent
 - 1.5.6. Understandable
 - 1.5.7. Modifiable

2 Organizational requirements

- 2.1 The process shall be applicable to the following types of projects at C Technologies.
 - 2.1.1 Release projects
 - 2.1.2 OEM Customizing projects
 - 2.1.3 New Development projects
 - 2.1.4 OEM Technical projects
- 2.2 The process shall be customized to fit C Technologies' project steering model.
- 2.3 The process shall be accepted and committed by all stakeholders at C Technologies affected by the requirement process.

3 Functional requirements

- 3.1. The process shall define requirements engineering actors.
 - 3.1.1. The process shall define responsibilities for the actors.
- 3.2. The process shall contain guidelines for how elicitation shall be performed.
 - 3.2.1. The process shall support continuous elicitation.
 - 3.2.2. The process shall include guidelines for arranged elicitation.
- 3.3. The process shall define which estimations to perform.
- 3.4. The process shall contain guidelines for how prioritization shall be performed.
- 3.5. It shall be possible to use the process in an iterative way.
- 3.6. Outputs from the process shall be defined.
- 3.7. The process shall define the following stakeholders' involvement.
 - 3.7.1 For Release projects the following stakeholders shall be involved: Customers such as, people already familiar with C Pen, new potential customers such as students, businessmen and other consumers. Marketing, Management, Purchasing, Development and Support.
 - 3.7.2 For OEM Customized projects the following stakeholders shall be involved: Customers such as companies that need special customized editions of C Technologies' already existing products. OEM Marketing, Management, Development and Purchasing.
 - 3.7.3 For New Development projects the following stakeholders shall be involved: Customers such as any consumer. Marketing, Management, Purchasing and Development.
 - 3.7.4 For OEM Technical projects the following stakeholders shall be involved: Customers such as companies that need C Technologies' technology for their own projects or products. OEM Marketing, Management, Purchasing and Development
- 3.8. The process shall be able to handle the following types of requirements:
 - 3.8.1. Functional requirements; are things the product must do.
 - 3.8.2. Non-functional requirements; are qualities the product must have.
 - 3.8.3. Usability requirements; are qualities regarding learnability and understandability.
- 3.9. The process shall describe how high-level requirements shall be documented.
- 3.10 The process shall support the following issues regarding traceability:
 - 3.10.1. It shall be possible to store all requirements.
 - 3.10.2. It shall be possible to trace the origin and history of a requirement.
 - 3.10.3. It shall be possible to determine when a requirement was issued.
 - 3.10.4. It shall be possible to put a requirement in a specific state.
 - 3.10.5. It shall be possible to change the state of a requirement.
 - 3.10.6. It shall be possible to determine to what project a certain requirement belongs.
 - 3.10.7. It shall be possible to save all elicited requirements.

Appendix D

Process proposal reference version

Appendix D presents a reference version of our process proposal.

Actors and their responsibilities

- **Issuer**
- **Requirements administrator**
- **Product board**
 - Product managers: is responsible for meetings and arranged elicitations.
 - Development manager
 - Innovation manager
 - Quality manager
 - Market representative
- **Project steering group**
 - Project Manager: has the overall responsibility and is responsible for meetings and arranged elicitations.
 - Management representative
 - Customer: All projects shall have a defined customer. Internal projects will have a product manager who shall be considered as customer.
 - Development representatives. Necessary for technical details.
 - Quality manager

Process phases

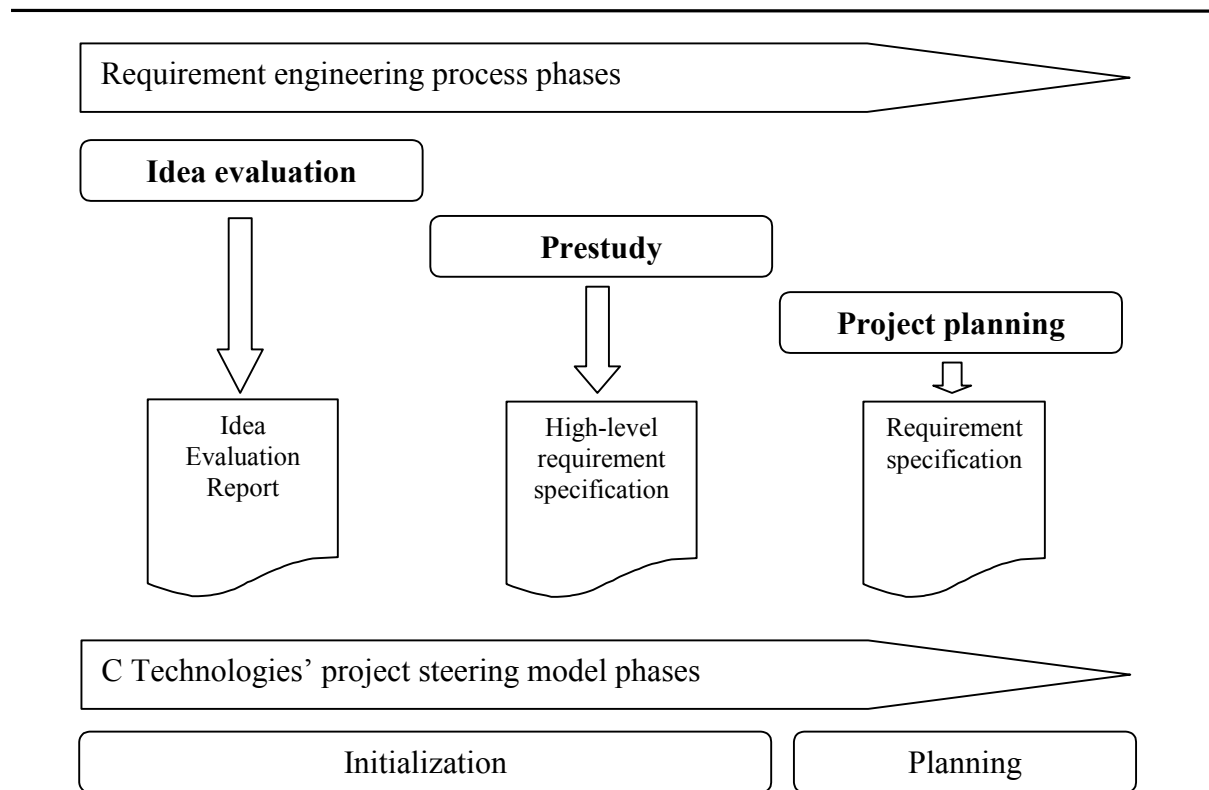


Figure 1. Requirement process phases and output

Idea evaluation phase

1. **Technical value**
 - New technology
 - Core technology
 - Patent issues
 - Etc
2. **Market value**
 - Market attraction
 - Market demand
 - Salability
 - Etc
3. **User benefit**
 - Usefulness for the user
 - Similar products from competitors
 - Etc

4. Development effort

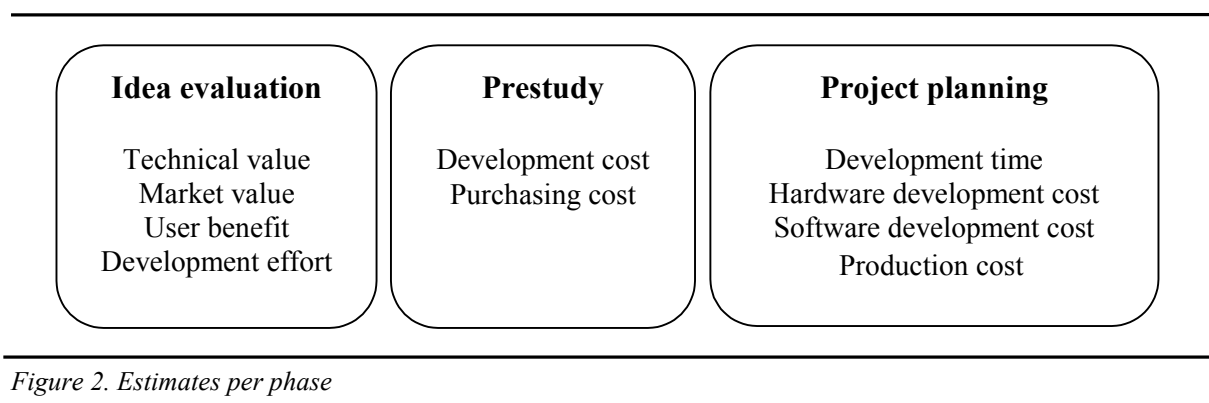
- Time
- Cost
- Resources
- Realization possibilities
- Etc

Prestudy phase

1. Development cost
2. Manufacturing cost (per unit)

Project planning phase

1. Development time
2. Hardware development costs
3. Software development costs
4. Production costs



Requirement states

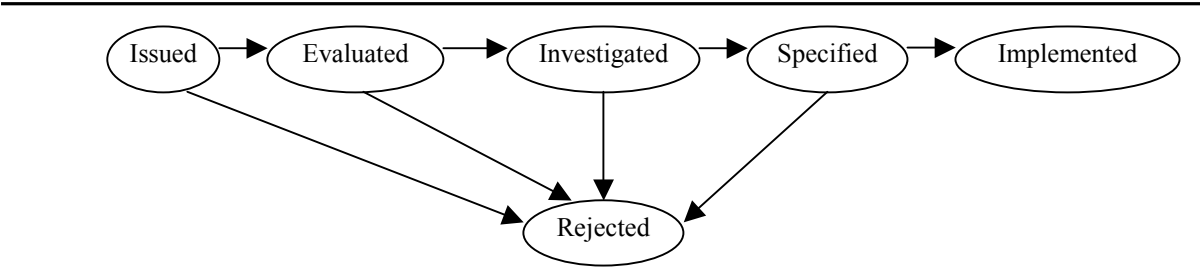


Figure 4.4. Requirements states in normal flow

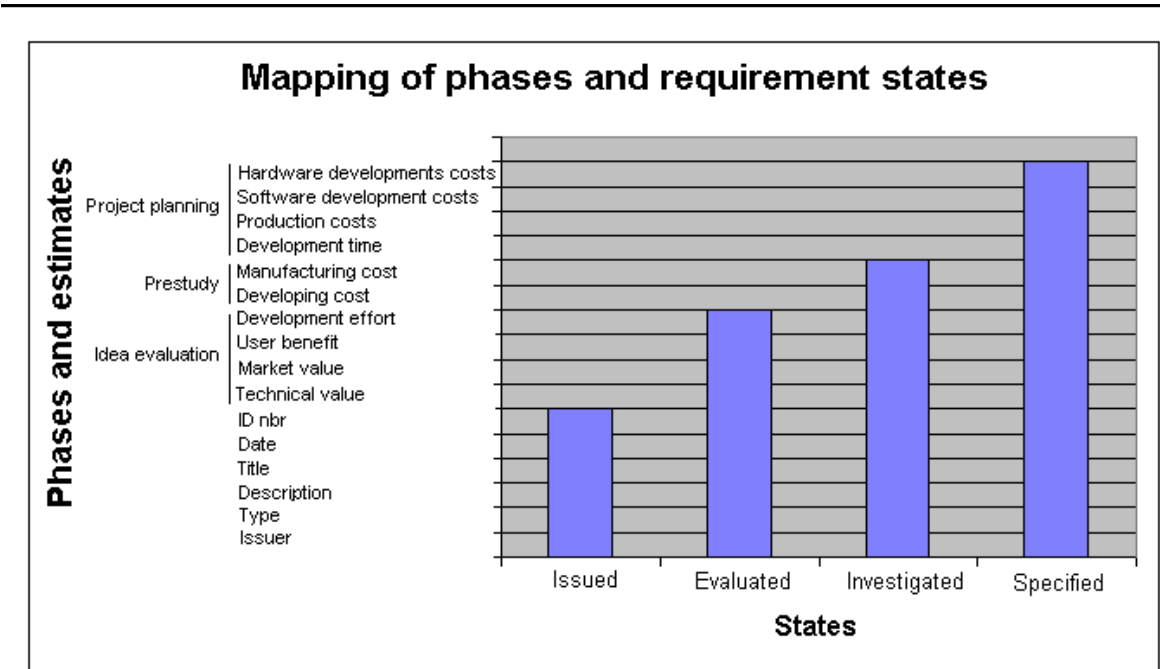


Figure 4.5 Mapping between phases, states and estimates

Process activities

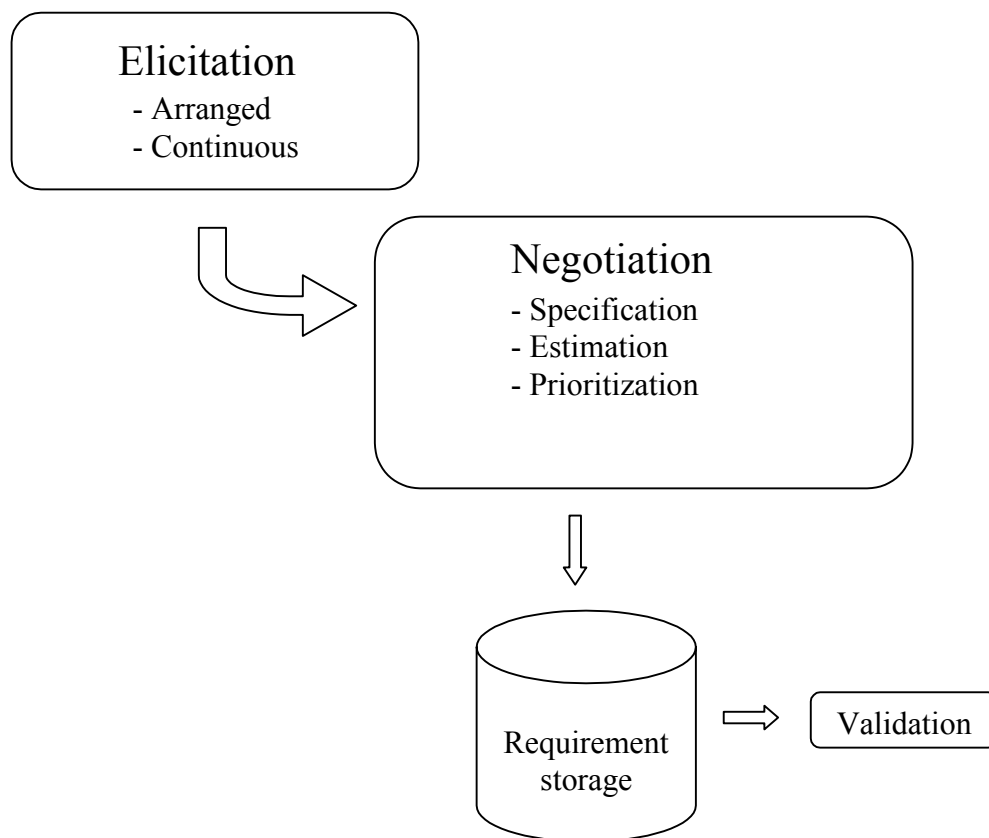


Figure 4.6. Requirement activities

Elicitation

Arranged elicitation

- Customer
- Support
- Development
- Project management

Continuous elicitation

- Customers and end users
- Employees at C Technologies

Specification

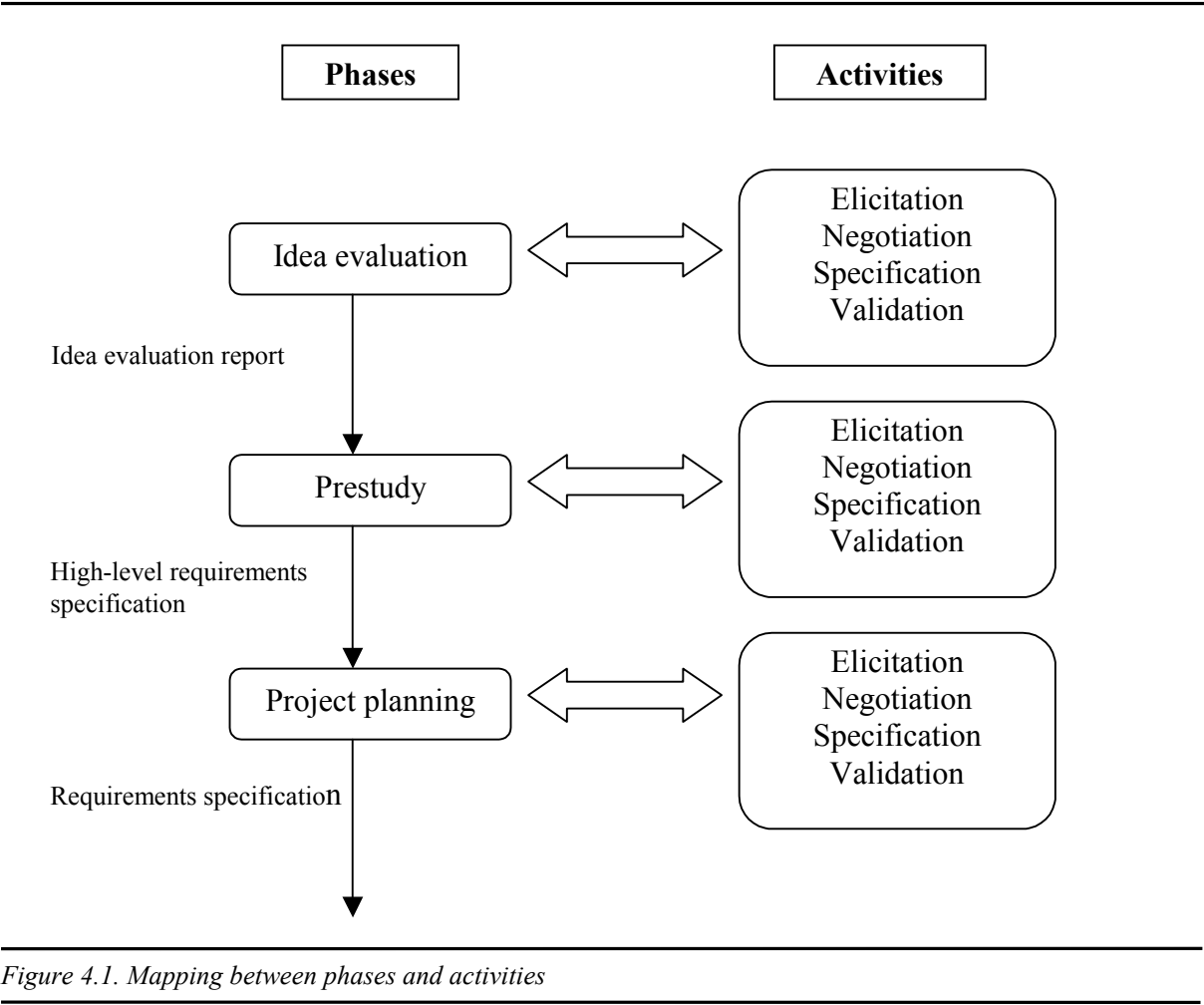
Negotiation

Estimation

Prioritization

- High Must be implemented
- Medium Implemented if time
- Low Next release or other product

Validation



Requirement attributes

- Unique identifier (number)
- Title
- Description
- Issuer email
- Issuing date
- Type (functional, nonfunctional, usability)
- State (issued, evaluated, investigated, specified, implemented, rejected)
- Estimates (technical value, market value, user benefit, development effort, development cost, manufacturing cost, development time, Hardware development costs, Software development costs, production costs)
- Priority (high, medium, low)
- Expire date
- Additional comments

Appendix E

Review checklist

Appendix E presents a specification review checklist example.

The specification as a whole must correspond to the following quality attributes:

Correctness	A requirements specification is correct if and only if every requirement stated therein represents something required of the system to be built.
Redundancy	A requirement may only exist once in the specification.
Completeness	Does the specification include all requirements that correspond to the customer's demands on the product? It must be investigated if any requirements are missing.
Consistency	There must be no contradictions between requirements.

Every requirement must correspond to the following quality attributes:

Understandable	Every requirement shall be easily read and understood
Verifiable/Testable	It shall be possible to verify that a requirement when implemented meets the specification. All requirements shall therefore be testable.
Unambiguous	It shall not be able to interpret a requirement in more than one way.

Appendix F

Database prototype

Appendix F describes our database prototype in more detail by showing the relations and attributes.

The database implementation

One idea with the database is that it shall be accessible for all stakeholders located at different places. Therefore we chose to make a web based interface. We used ASP (Active Server Pages), actually Visual Basic script together with html and connected to a web server running an SQL server 7.0 database.

The work with the database started with the producing of an Entity/Relationship model [Connolly et al, 1999]. The model helped us define the different entities, relations and attributes needed. The model, see figure 4.1, its attributes and relations are explained below.

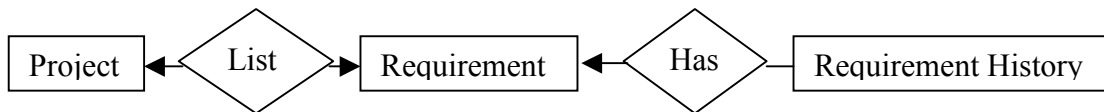


Figure 4.1 Entity/Relationship model for our prototype

The relations and attributes are presented below. Some of the attributes have limited value alternatives, which also are presented. The primary key for respective relation is underlined and the foreign keys are written in italics.

Project (IdNbr, Type, Name, Responsible)

“Type” alternatives: IdeaEvaluation, PreStudy, Project

Requirement (IdNbr, Type, Title, Description, IssuerEmail, Date, State, CustomerNeed, TechniqueLevel, Cost, Salability, Priority, *Parent*, Comments)

“Type” alternatives: Functional, Non-functional, Usability

“State” alternatives: New, Approved, Estimated, Planned, Implemented, Rejected

RequirementProjectList (ReqId, ProjectId)

RequirementHistory (IdNbr, *ReqIdNbr*, Type, Title, Description, IssuerEmail, Date, State, CustomerNeed, TechniqueLevel, Cost, Salability, Priority, *Parent*, Comments)

Each project shall be able to contain several requirements and each requirement shall be able to belong to several different projects. The history (changes) from all attributes belonging to a specific requirement shall be saved. Each requirement can be updated several times, and by that use several history fields.

Appendix G

Prototype implementation code

Appendix G presents all code produced for the web forms, both the html code and the visual basic script code.

In order to run the prototype a computer that is capable of handling Active Server Pages (ASP) must be used. This approach means that all code is executed on the server and that the web browser used can be of any type or version. The computer running the prototype must also be running as a web server. The web server installation depends on what operating system the computer is running. Often a web server installation enables .asp files to be executed.

The .asp files shall be located in the directory named wwwroot located under c:\inetpub\. This directory is automatically created when the web server is installed. To view the web pages and run the prototype open a web browser and type localhost in the address field.

The database according to appendix F must also be in place. Projects must be manually added to the database when created. The projects must be:

<u>IdNbr</u>	<u>Name</u>
1	cpen800
2	cpen600
3	oscar
4	hugin
5	gleipner
6	new
7	bodn

The attributes Type and Responsible are not used and may be left out.

default.asp

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Requirement management tool</title>
</head>
<body>
<p><b><font size="4">Requirements management service for C Technologies
AB</font></b></p>
<hr>
<p>This is the starting point for requirements management. With this prototype
it is possible to:</p>
<ul>
<li><a href="hlreqcollection.asp">Submit a requirement or an idea</a></li>
</ul>
<ul>
<li><a href="search.asp">Search function<br>
</a>(Search a requirement and update, brake down or view its history)</li>
</ul>
<hr>
<p><font size="1">Prototype produced by <a href="mailto:cdictionary4@cpen.com">
Urban Martinsson and Åsa
Karlsson</a> as part of our Bachelor Thesis<br>
Last updated <!--webbot bot="Timestamp" s-type="EDITED"
s-format="%Y-%m-%d" --> </font> </p>
</body>
</html>
```

hlreqcollection.asp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<TITLE>Requirement Collection Form</TITLE>
<!-- #include virtual="adovbs.inc" -->
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<META content="Microsoft FrontPage 4.0" name=GENERATOR></HEAD>
<BODY>
<%
    if request.form("action") = "Submit requirement" Then
        set mdb = server.createobject("ADODB.Connection")
        mdb.open "PROVIDER=SQLOLEDB;DATA
SOURCE=192.168.2.192;UID=guran;PASSWORD=urban;DATABASE=RHandler"
        Product = request.form("Product")
        Titel = request.form("Title")
        Description = request.form("Description")
        Typ = request.form("Typ")
        Submitteremail = request.form("SubmitterEmail")
        Comments = request.form("Comments")
        SQLQuery = "INSERT INTO Req(Typ, Title, Description, IssuerEmail, Date,
State, Comments) VALUES(' & Typ & ', ' & Titel & ', ' & Description & ', ' &
Submitteremail & ', ' & Date & ', 'Issued', ' & Comments & ');"
        set MyRs = mdb.execute(SQLQuery)
        Set RecSet = Server.CreateObject("ADODB.Recordset")
        SQLReqnr = "SELECT MAX(idnbr) AS reqnumber FROM req"
        RecSet.Open SQLReqnr, mdb, adOpenStatic, adLockOptimistic
        requirementnr = RecSet("reqnumber")
        if Product = "800C" Then
            project = 1
        elseif Product = "600C" then
            project = 2
        elseif Product = "Oscar" then
            project = 3
```

```

        elseif Product = "Hugin" then
            project = 4
        elseif Product = "Gleipner" then
            project = 5
        elseif Product = "New" then
            project = 6
        elseif Product = "Bodn" then
            project = 7
        end if
        SQLQuery = "INSERT INTO list(project, req) VALUES('" & project & "', '" &
requirementnr & "');"
        set MyRs = mdb.execute(SQLQuery)
        mdb.close
    end if
%>

<H1><font size="4"><b>Requirement Collection Form (prototype)</b></font></H1>
<HR>
<FORM method="post">
<TABLE cellSpacing=5 width="101%" border=0 height="382">
    <TBODY>
        <TR>
            <TD width="20%" height="24"><STRONG>Product:</STRONG></TD>
            <TD width="36%" height="24"><SELECT size=1 name=Product> <OPTION value=800C
selected>C-Pen 800C</OPTION> <OPTION value=600C>C-Pen 600C</OPTION>
<OPTION value=Gleipner>Gleipner</OPTION> <OPTION
value=Hugin>Hugin</OPTION> <OPTION value=Oscar>Oscar</OPTION><OPTION
value=Bodn>Bodn</OPTION> <OPTION
value=New>New</OPTION></SELECT></TD>
            <TD width="4%" height="24"></TD>
            <TD width="46%" height="24">What product does this requirement mainly affect?<%
=project %></TD></TR>
        <TR>
            <TD width="20%" height="25"><STRONG>Title: </STRONG></TD>
            <TD width="36%" height="25"><INPUT size=42 name=Title></TD>
            <TD width="4%" height="25"></TD>
            <TD width="46%" height="25">A one line title that describes the
requirement</TD></TR>
        <TR>
            <TD width="20%" height="119"><STRONG>Description: </STRONG></TD>
            <TD width="36%" height="119"><TEXTAREA name=Description rows=5
cols=36></TEXTAREA></TD>
            <TD width="4%" height="119"></TD>
            <TD width="46%" height="119">A detailed description of the
requirement</TD></TR>
        <tr>
            <TD width="20%" height="25"><strong>Type:</strong></TD>
            <TD width="36%" height="25"><B><SELECT size=1 name=Typ>
<OPTION value="Functional" selected>Functional</OPTION><OPTION value="Non-
functional">Non-functional</OPTION></SELECT> </B></TD>
            <TD width="4%" height="25"></TD>
            <TD width="46%" height="25">Declare what type this requirement is</TD>
        </tr>
        <tr>
            <!-- här skall det läsas in alla krav från databasen tillhörande ett visst
projekt -->
        </tr>
        <tr>
            <TD width="20%" height="25"><STRONG>Submitters e-mail:</STRONG></TD>
            <TD width="36%" height="25"><INPUT size=42 name=SubmitterEmail></TD>
            <TD width="4%" height="25"></TD>
            <TD width="46%" height="25">Add submitters (your) e-mail</TD>
        </tr>
        <tr>
            <TD width="20%" height="81"><STRONG>Comments:</STRONG></TD>
            <TD width="36%" height="81"><TEXTAREA name=Comments cols=36
rows="3"></TEXTAREA></TD>

```

[illegible]

```
<html>
<body>
<h1><font size="4"><b>Requirements&nbsp;search</b></font></h1>
<hr>
<p>This search function makes it possible to:</p>
<ul>
  <li><a href="show1.asp?requiemet=all">Search all requirements in the
database</a> </li>
  <li><a href="listreqstate1.html">Search requirements with a specific state
belonging to a specific project</a></li>
</ul>
<hr>
<ul>
  <li><a href="hlreqcollection.asp">Submit a requirement or an idea</a></li>
</ul>
  <li><a href="default.asp">Requirement Service start point</a></li>
</ul>
<p><font size="1">Prototype produced by <a href="mailto:cdictionary4@cpen.com">
Urban Martinsson and Åsa
Karlsson</a> as part of our Bachelor Thesis<br>
Last updated <!--webbot bot="Timestamp" s-type="EDITED"
s-format="%Y-%m-%d" --> </font> </p>
</body>
</html>
```

show1.asp

```
<html>
<!-- #include virtual="adovbs.inc" -->
<% set Connect = server.createobject("ADODB.Connection")
    Connect.open "PROVIDER=SQLOLEDB;DATA
SOURCE=192.168.2.192;UID=guran;PASSWORD=urban;DATABASE=RHandler"

    SQLQuery = "SELECT Req.* FROM Req"
    Set RecSet = Connect.Execute(SQLQuery)
%>
<h1>
<font size="4"><b>List of all requirements in the database</b></font>
</h1>
<hr width="1698">
<table border="1" width="1728">
    <tr>
        <td width="80" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Project</b></td>
        <td width="35" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Id</b></td>
        <td width="163" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Title</b></td>
        <td width="145" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Description</b></td>
        <td width="90" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Type</b></td>
        <td width="80" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>State</b></td>
        <td width="61" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Cost
estimate</b></td>
        <td width="78" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Technique
level</b></td>
        <td width="73" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Salability</b></td>
        <td width="76" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Customer
need</b></td>
        <td width="57" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Priority</b></td>
        <td width="52" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Parent</b></td>
        <td width="152" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Additional comments</b></td>
        <td width="108" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Date</b></td>
        <td width="194" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Issuer
Email</b></td>
        <td width="59" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Update</b></td>
        <td width="47" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Break
down</b></td>
        <td width="80" bordercolor="#FFFFFF" valign="bottom" align="left"><b>View
history</b></td>
    </tr>
</table>

<% Do Until RecSet.EOF
    reqId = RecSet("IdNbr")
    SQLQuery1 = "SELECT project.Name FROM project, list WHERE list.req = ' & reqId
& ' AND list.project = project.idNbr"
    Set RecSet1 = Connect.Execute(SQLQuery1)
%>

<table border="1" width="1728">
    <tr>
        <td width="80" bordercolor="#FFFFFF" align="left"><% =RecSet1("Name") %></td>
        <td width="35" bordercolor="#FFFFFF" align="left"><% =RecSet("IdNbr") %></td>
        <td width="163" bordercolor="#FFFFFF" align="left"><% =RecSet("Title") %></td>
```

```

        <td width="146" bordercolor="#FFFFFF" align="left"><% =RecSet("Description")
%></td>
        <td width="90" bordercolor="#FFFFFF" align="left"><% =RecSet("Typ") %></td>
        <td width="80" bordercolor="#FFFFFF" align="left"><% =RecSet("State") %></td>
        <td width="61" bordercolor="#FFFFFF" align="left"><% =RecSet("RCost") %></td>
        <td width="78" bordercolor="#FFFFFF" align="left"><% =RecSet("RTechniquelevel")
%></td>
        <td width="73" bordercolor="#FFFFFF" align="left"><% =RecSet("RSaleability")
%></td>
        <td width="76" bordercolor="#FFFFFF" align="left"><% =RecSet("RCustomerneed")
%></td>
        <td width="57" bordercolor="#FFFFFF" align="left"><% =RecSet("priority")
%></td>
        <td width="52" bordercolor="#FFFFFF" align="left"><% =RecSet("Father") %></td>
        <td width="152" bordercolor="#FFFFFF" align="left"><% =RecSet("comments")
%></td>
        <td width="108" bordercolor="#FFFFFF" align="left"><% =RecSet("Date") %></td>
        <td width="194" bordercolor="#FFFFFF" align="left"><a href="mailto:<%
=RecSet("IssuerEmail") %>"><% =RecSet("IssuerEmail") %></td>
        <td width="59" bordercolor="#FFFFFF" align="left"><a
href="modification.asp?requirement=<% =RecSet("IdNbr") %>">Update</a>&nbsp;</td>
        <td width="47" bordercolor="#FFFFFF" align="left"><a
href="split2.asp?requirement=<% =RecSet("IdNbr") %>">Break down</a>&nbsp;</td>
        <td width="80" bordercolor="#FFFFFF" align="left"><a
href="ShowHistory.asp?requirement=<% =RecSet("IdNbr") %>">View
history</a>&nbsp;</td>
    </tr>
</table>

<% RecSet.MoveNext
    Loop
    Connect.Close
%>

<hr width="1698">
<ul>
<li><a href="listreq.asp">Chose other project</a></li>
</ul>
<ul>
<li><a href="hlreqcollection.asp">Submit a requirement or an idea</a></li>
<li><a href="search.asp">Search function</a> (Update, Break down or View history
for selected requirement)</li>
</ul>
<ul>
<li><a href="default.asp">Requirement Service start point</a></li>
</ul>
<p><font size="1">Prototype produced by <a href="mailto:cdictionary4@cpen.com">
Urban Martinsson and Åsa
Karlsson</a> as part of our Bachelor Thesis<br>
    Last updated <!--webbot bot="Timestamp" s-type="EDITED"
        s-format="%Y-%m-%d" --> </font></p>
</html>

```

listreqstate1.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<TITLE>Requirement Collection Form</TITLE>
<!-- #include virtual="adovbs.inc" -->
<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<META content="Microsoft FrontPage 4.0" name=GENERATOR></HEAD>
<BODY>

<H1><font size="4"><b>Search requirements belonging to a specific project with a
specific
state</b></h1>

```



```
<HR>  
<FORM method="post" name="action" action="show2.asp">  
<TABLE cellSpacing=5 width="101%" border=0 height="145">  
  <TBODY>  
    <TR>  
      <TD width="10%" height="1"><STRONG>Select project:</STRONG></TD>  
      <TD width="51%" height="1">  
        <SELECT size=1 name=Project>  
          <OPTION value=800C selected>C-Pen 800C</OPTION>  
          <OPTION value=600C>C-Pen 600C</OPTION>  
          <OPTION value=Gleipner>Gleipner</OPTION>  
          <OPTION value=Hugin>Hugin</OPTION>  
          <OPTION value=Oscar>Oscar</OPTION>  
          <OPTION value=Bodn>Bodn</OPTION>  
        </SELECT></TD>  
    </TR>  
    <tr>  
      <TD width="10%" height="1"><strong>Select state:</strong></TD>  
      <TD width="51%" height="1"><B>  
        <SELECT size=1 name=State>  
          <option selected value="All">All states</option>  
          <option>-----</option>  
          <OPTION value="Issued">Issued</OPTION>  
          <OPTION value="Evaluated">Evaluated</OPTION>  
          <OPTION value="Investigated">Investigated</OPTION>  
          <option value="Specified">Specified</option>  
          <option value="Implemented">Implemented</option>  
          <option value="Rejected">Rejected</option>  
        </SELECT> </B></TD>  
    </tr>  
    <TR>  
      <TD width="10%" height="15"></TD>  
      <TD width="51%" height="15">  
        <INPUT TYPE="submit" NAME="action"  
VALUE="Search">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        </TD>  
    </TR></TBODY></TABLE>  
</form>  
<HR>  
</font>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
<ul>  
  <li><a href="hlreqcollection.asp">Submit a requirement or an idea</a></li>  
  <li><a href="search.asp">Search function</a> (Update, Break down or View history  
for selected requirement)</li>  
</ul>  
<ul>  
  <li><a href="default.asp">Requirement Service start point</a></li>  
</ul>  
<p><font size="1">Prototype produced by <a href="mailto:cdictionary4@cpen.com">  
Urban Martinsson and Åsa  
Karlsson</a> as part of our Bachelor Thesis<br>  
Last updated <!--webbot bot="Timestamp" s-type="EDITED"  
s-format="%Y-%m-%d" --> </font></p>  
</BODY>  
</HTML>
```

```
<html>
<!-- #include virtual="adovbs.inc" -->
<% set Connect = server.createobject("ADODB.Connection")
    Connect.open "PROVIDER=SQLOLEDB;DATA
SOURCE=192.168.2.192;UID=quran;PASSWORD=urban;DATABASE=RHandler"
```

```

    If Request.Form("Project") = "800C" OR Request.Form("project")= "600C" OR
Request.Form("project")= "Oscar" OR Request.Form("project")= "Hugin" OR
Request.Form("project")= "Gleipner" OR Request.Form("project")= "Bodn" Then

        Project = Request.Form("Project")
        state = Request.Form("State")

        If Project = "800C" Then
            product = "cpen800"

        ElseIf Project = "600C" Then
            product = "cpen600"

        ElseIf Project = "Oscar" Then
            product = "oscar"

        ElseIf Project = "Hugin" Then
            product = "hugin"

        ElseIf Project = "Gleipner" Then
            product = "gleipner"

        ElseIf Project = "Bodn" Then
            product = "bodn"

        End If

        Set RecSet = Server.CreateObject("ADODB.Recordset")

        If state = "All" Then
            Requirement = "SELECT req.* FROM req, project, list WHERE project.name =
'" & product & "' AND project.IdNbr = List.project AND List.req = Req.IdNbr"

            ElseIf Project = "800C" Then
                Requirement = "SELECT req.* FROM req, project, list WHERE project.name =
'cpen800' AND project.IdNbr = List.project AND List.req = Req.IdNbr AND Req.state =
'" & state & "'"

            ElseIf Project = "600C" Then
                Requirement = "SELECT req.* FROM req, project, list WHERE project.name =
'cpen600' AND project.IdNbr = List.project AND List.req = Req.IdNbr AND Req.state =
'" & state & "'"

            ElseIf Project = "Oscar" Then
                Requirement = "SELECT req.* FROM req, project, list WHERE project.name =
'oscar' AND project.IdNbr = List.project AND List.req = Req.IdNbr AND Req.state =
'" & state & "'"

            ElseIf Project = "Hugin" Then
                Requirement = "SELECT req.* FROM req, project, list WHERE project.name =
'hugin' AND project.IdNbr = List.project AND List.req = Req.IdNbr AND Req.state =
'" & state & "'"

            ElseIf Project = "Gleipner" Then
                Requirement = "SELECT req.* FROM req, project, list WHERE project.name =
'gleipner' AND project.IdNbr = List.project AND List.req = Req.IdNbr AND Req.state
= '" & state & "'"

            ElseIf Project = "Bodn" Then
                Requirement = "SELECT req.* FROM req, project, list WHERE project.name =
'bodn' AND project.IdNbr = List.project AND List.req = Req.IdNbr AND Req.state = '"
& state & "'"

        End If
        RecSet.Open Requirement, Connect, adOpenStatic, adLockOptimistic
%>

```

<h1>

```

<font size="4"><b>List of requirements belonging to a specific project with a
specific state
</b></font>
</h1>
<hr width="1648">
<table border="1" width="1648">
  <tr>
    <td width="35" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Id</b></td>
    <td width="163" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Title</b></td>
    <td width="145" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Description</b></td>
    <td width="90" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Type</b></td>
    <td width="80" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>State</b></td>
    <td width="61" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Cost
estimate</b></td>
    <td width="78" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Technique
level</b></td>
    <td width="73" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Salability</b></td>
    <td width="76" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Customer
need</b></td>
    <td width="57" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Priority</b></td>
    <td width="52" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Parent</b></td>
    <td width="152" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Additional comments</b></td>
    <td width="108" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Date</b></td>
    <td width="194" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Issuer
Email</b></td>
    <td width="59" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Update</b></td>
    <td width="47" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Break
down</b></td>
    <td width="80" bordercolor="#FFFFFF" valign="bottom" align="left"><b>View
history</b></td>
  </tr>
</table>

<% Do Until RecSet.EOF %>

<table border="1" width="1648">
  <tr>
    <td width="35" bordercolor="#FFFFFF" align="left"><% =RecSet("IdNbr") %></td>
    <td width="163" bordercolor="#FFFFFF" align="left"><% =RecSet("Title") %></td>
    <td width="146" bordercolor="#FFFFFF" align="left"><% =RecSet("Description")
%></td>
    <td width="90" bordercolor="#FFFFFF" align="left"><% =RecSet("Typ") %></td>
    <td width="80" bordercolor="#FFFFFF" align="left"><% =RecSet("State") %></td>
    <td width="61" bordercolor="#FFFFFF" align="left"><% =RecSet("RCost") %></td>
    <td width="78" bordercolor="#FFFFFF" align="left"><% =RecSet("RTechniquelevel")
%></td>
    <td width="73" bordercolor="#FFFFFF" align="left"><% =RecSet("RSaleability")
%></td>
    <td width="76" bordercolor="#FFFFFF" align="left"><% =RecSet("RCustomerneed")
%></td>
    <td width="57" bordercolor="#FFFFFF" align="left"><% =RecSet("priority")
%></td>
    <td width="52" bordercolor="#FFFFFF" align="left"><% =RecSet("Father") %></td>
    <td width="152" bordercolor="#FFFFFF" align="left"><% =RecSet("comments")
%></td>
    <td width="108" bordercolor="#FFFFFF" align="left"><% =RecSet("Date") %></td>

```



```

Priority = RecSet("Priority")
CustomerNeed = RecSet("RCustomerNeed")
TechniqueLevel = RecSet("RTechniqueLevel")
Cost = RecSet("RCost")
Salability = RecSet("RSaleability")
Parent = RecSet("Father")
RecSet.close

SQLQuery = "SELECT Project.IdNbr, Project.Name FROM project, list WHERE list.req
= '" & reqId & "' AND list.project = project.IdNbr"
RecSet.Open SQLQuery, Connect, adOpenStatic, adLockOptimistic

projectNbr = RecSet("IdNbr")
projectName = RecSet("Name")
RecSet.Close

If request.form("action") = "Update requirement" Then

    'Skicka alla attribut som finns till historytabellen
    SQLQuery = "INSERT INTO ReqHistory(reqid, Typ, Title, Description,
IssuerEmail, Date, State, RCustomerneed, RTechniquelevel, RCost, RSalability,
Priority, Father, Comments) VALUES('" & reqId & "', '" & Typ & "', '" & Titel & "',
'" & Description & "', '" & Submitteremail & "', '" & Datum & "', '" & State & "',
'" & CustomerNeed & "', '" & TechniqueLevel & "', '" & Cost & "', '" & Salability &
"', '" & Priority & "', '" & Parent & "', '" & Comments & ');"
    set RecSet = Connect.execute(SQLQuery)

    'skicka all ny information till samma värde i databasen med update, all
    gammal info skall ersättas med ny (UPDATE)

    newTitel = Request.form("Title")
    newDescription = Request.form("Description")
    newTyp = Request.form("Typ")
    newState = Request.form("State")
    newSubmitteremail = Request.form("Submitteremail")
    newComments = Request.form("Comments")
    newDatum = Date
    newPriority = Request.form("Priority")
    newCustomerNeed = Request.form("R4")
    newTechniqueLevel = Request.form("R2")
    newCost = Request.form("R1")
    newSalability = Request.form("R3")

    SQLQuery = "UPDATE Req SET Req.RCost = '" & newCost & "', Req.Title = '" &
newTitel & "', Req.Description = '" & newDescription & "', Req.Typ = '" & newTyp &
"', Req.IssuerEmail = '" & newSubmitteremail & "', Req.RTechniqueLevel = '" &
newTechniquelevel & "', Req.RSaleability = '" & newSalability & "', Req.Priority =
'" & newPriority & "', Req.Comments = '" & newComments & "', Req.Date = '" &
newDatum & "', Req.State = '" & newState & "', Req.RCustomerNeed = '" &
newCustomerneed & "'" WHERE Req.IdNbr = '" & reqId & "'"

    set RecSet = connect.execute(SQLQuery)

    'lagrar undan nya värden för presentation i fälten på skärmen
    Titel = newTitel
    Description = newDescription
    Typ = newTyp
    State = newState
    Submitteremail = newSubmitteremail
    Comments = newComments
    Datum = newDatum
    Priority = newPriority
    CustomerNeed = newCustomerNeed
    TechniqueLevel = newTechniqueLevel
    Cost = newCost
    Salability = newSalability

```

```

End If
Connect.close
%>

<H1><font size="4"><b>Requirement Update Form</b></font></H1>
<HR>
<FORM method="post" name="1">
<INPUT type="hidden" value=New_requirement
name=Subject><INPUT type="hidden
value=pbe@cpen.com name=Recipient>
<TABLE cellSpacing=5 width="101%" border=0 height="373">
  <TBODY>
    <TR>
      <TD width="20%" height="21"><STRONG>Project:</STRONG></TD>
      <TD width="28%" height="21"><% =projectName %></TD>
      <TD width="3%" height="21"></TD>
      <TD width="55%" height="21"> The projects this requirement is connected to
</TD></TR>
    <TR>
      <TD width="20%" height="21"><strong> Id:</strong></TD>
      <TD width="28%" height="21"><% =reqId %></TD>
      <TD width="3%" height="21"></TD>
      <TD width="55%" height="21"> This requirement identifier </TD></TR>
    <TR>
      <TD width="20%" height="25"><strong>Title:</strong></TD>
      <TD width="28%" height="25"><INPUT type="text" size=42 name=Title value="<%
=Titel %>"></TD>
      <TD width="3%" height="25"></TD>
      <TD width="55%" height="25">If necessary, specify the title more</TD></TR>
    <TR>
      <TD width="20%" height="119"><STRONG>Description: </STRONG></TD>
      <TD width="28%" height="119"><TEXTAREA name=Description rows=5 cols=36><%
=Description %>
</TEXTAREA></TD>
      <TD width="3%" height="119"></TD>
      <TD width="55%" height="119">If necessary, specify the description
more</TD></TR>
    <tr>
      <TD width="20%" height="25"><strong>Type:</strong></TD>
      <TD width="36%" height="25"><b><SELECT size=1 name=Typ>
<OPTION value="<% =Typ %>" selected><% =Typ %></OPTION>
<OPTION value="Functional">Functional</OPTION><OPTION value="Non-
functional">Non-functional</OPTION><OPTION
value="Usability">Usability</OPTION></SELECT> </b></TD>
      <TD width="3%" height="25"></TD>
      <TD width="55%" height="25">Declare what type this requirement is</TD>
</tr>
    <tr>
      <TD width="20%" height="28"><strong> State:</strong></TD>
      <TD width="28%" height="28"><b><SELECT size=1 name=State>
<OPTION value="<% =State %>" selected><% =State %></OPTION>
<OPTION value="New">Issued</OPTION>
<option>Evaluated</option>
<option>Investigated</option>
<option>Specified</option>
<option>Implemented</option>
<option>Rejected</option>
</SELECT> </b></TD>
      <TD width="3%" height="28"></TD>
      <TD width="55%" height="28">Declare what state this requirement has</TD>
</tr>
    <tr>
      <TD width="20%" height="25" valign="top"><b>Development effort:</b></TD>
      <TD width="28%" height="25" valign="top">
<b><SELECT size=1 name=R1><OPTION
value="<% =Cost %>" selected><% =Cost %></OPTION>
<option>1</option>
<option>2</option>

```

```

        <option>3</option>
        <option>4</option>
        <option>5</option>
        <option>6</option>
        <option>7</option>
        <option>8</option>
        </SELECT> </B></TD>
    <TD width="3%" height="25"></TD>
    <TD width="55%" height="25">Specify the development effort by selecting a
        number</TD>
</tr>
<tr>
    <TD width="20%" height="25" valign="top"><b>Technical value:</b></TD>
    <TD width="28%" height="25" valign="top"><B><SELECT size=1 name=R2><OPTION
        value="<% =TechniqueLevel %>" selected><% =TechniqueLevel %></OPTION>
        <option>1</option>
        <option>2</option>
        <option>3</option>
        <option>4</option>
        <option>5</option>
        <option>6</option>
        <option>7</option>
        <option>8</option>
        </SELECT> </B></TD>
    <TD width="3%" height="25"></TD>
    <TD width="55%" height="25">Specify the technical value by selecting a
        number</TD>
</tr>
<tr>
    <TD width="20%" height="25" valign="top"><b>Market value:</b></TD>
    <TD width="28%" height="25" valign="top"><B><SELECT size=1 name=R3><OPTION
        value="<% =Salability %>" selected><% =Salability %></OPTION>
        <option>1</option>
        <option>2</option>
        <option>3</option>
        <option>4</option>
        <option>5</option>
        <option>6</option>
        <option>7</option>
        <option>8</option>
        </SELECT> </B></TD>
    <TD width="3%" height="25"></TD>
    <TD width="55%" height="25">Specify the market value by selecting a number</TD>
</tr>
<tr>
    <TD width="20%" height="25" valign="top"><b>User benefit:</b></TD>
    <TD width="28%" height="25" valign="top"><B><SELECT size=1 name=R4><OPTION
        value="<% =CustomerNeed %>" selected><% =CustomerNeed %></OPTION>
        <option>1</option>
        <option>2</option>
        <option>3</option>
        <option>4</option>
        <option>5</option>
        <option>6</option>
        <option>7</option>
        <option>8</option>
        </SELECT> </B></TD>
    <TD width="3%" height="25"></TD>
    <TD width="55%" height="25">Specify the user benefit by selecting a number</TD>
</tr>
<tr>
    <TD width="20%" height="25"><b>Priority:</b></TD>
    <TD width="28%" height="25"><B><SELECT size=1 name=Priority><OPTION
        value="<% =Priority %>" selected><% =Priority %></OPTION>
        <option>1</option>
        <option>2</option>
        <option>3</option>
        </SELECT> </B></TD>

```

```
<TD width="3%" height="25"></TD>  
<TD width="55%" height="25">Declare what priority the requirement has</TD>  
</tr>  
<tr>  
    <TD width="20%" height="25"><b>Parent:</b></TD>  
    <TD width="28%" height="25"><% =Parent %></TD>  
    <TD width="3%" height="25"></TD>  
    <TD width="55%" height="25">The requirement this requirement originate  
from</TD>  
</tr>  
<tr>  
    <TD width="20%" height="25"><b>Original issuing date:</b></TD>  
    <TD width="28%" height="25"><% =Datum %></TD>  
    <TD width="3%" height="25"></TD>  
    <TD width="55%" height="25">The date when this requirement was last  
updated&nbsp;&nbsp;&nbsp;</TD>  
</tr>  
<tr>  
    <TD width="20%" height="25"><strong>Editors e-mail:</strong></TD>  
    <TD width="28%" height="25"><INPUT size=42 name=Submitteremail value="<%  
=Submitteremail %>"></TD>  
    <TD width="3%" height="25"></TD>  
    <TD width="55%" height="25">Update this field with the editors (your) e-  
mail</TD>  
</tr>  
<tr>  
    <TD width="20%" height="85"><STRONG>Comments:</STRONG></TD>  
    <TD width="28%" height="85"><TEXTAREA name=Comments cols=36 rows="3"><%  
=Comments %></TEXTAREA></TD>  
    <TD width="3%" height="85"></TD>  
    <TD width="55%" height="85">Any other comments you want to make in order to  
promote  
this requirement. For example why this requirement should be implemented  
or what problem this requirement is trying to solve.</TD>  
</tr>  
<TR>  
    <TD width="20%" height="45"></TD>  
    <TD width="28%" height="45"><INPUT TYPE="submit" NAME="action" VALUE="Update  
requirement">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <INPUT type=reset value=Reset name=B2></TD>  
    <TD width="3%" height="45"></TD>  
    <TD width="55%" height="45"></TD></TR></TBODY></TABLE>  
<DIV align=center>  
</DIV> </FORM>  
<hr>  
<ul>  
    <li><a href="hlreqcollection.asp">Submit a requirement or an idea</a></li>  
    <li><a href="listreq.asp">List requirements for a specific project</a>  
(Update, Break down or View history for selected requirement)  
</li>  
</ul>  
<ul>  
    <li><a href="default.asp">Requirement Service start point</a></li>  
</ul>  
<p><font size="1">Prototype produced by <a href="mailto:cdictionary4@cpen.com">  
Urban Martinsson and Åsa  
Karlsson</a> as part of our Bachelor Thesis<br>  
Last updated <!--webbot bot="Timestamp" s-type="EDITED"  
s-format="%Y-%m-%d" --> </font> </p>  
</BODY></HTML>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD>
<TITLE>Requirement Collection Form</TITLE>
```



```

<META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
<META content="Microsoft FrontPage 4.0" name=GENERATOR></HEAD>
<!-- #include virtual="adovbs.inc" -->
<BODY>

<%
    set Connect = server.createObject("ADODB.Connection")
    Connect.open "PROVIDER=SQLOLEDB;DATA
SOURCE=192.168.2.192;UID=guran;PASSWORD=urban;DATABASE=RHandler"

    'tag reda på föräldern/ursprungskravet på högre nivå
    Parent = Request.QueryString("requirement")

    Set RecSet = Server.CreateObject("ADODB.Recordset")
    SQLQuery1 = "SELECT Project.IdNbr, project.Name FROM project, list WHERE
list.req = '" & Parent & "' AND list.project = project.IdNbr"
    RecSet.Open SQLQuery1, Connect, adOpenStatic, adLockOptimistic

    projectNbr = RecSet("IdNbr")
    projectName = RecSet("Name")
    RecSet.Close

    Set RecSet = Server.CreateObject("ADODB.Recordset")
    SQLQuery2 = "SELECT req.title FROM req WHERE req.IdNbr = '" & Parent & "'"
    RecSet.Open SQLQuery2, Connect, adOpenStatic, adLockOptimistic

    reqtitle = RecSet("Title")

    If request.form("action") = "Submit requirement" Then

        'Titel = reqtitle
        Description = request.form("Description")
        Typ = request.form("Typ")
        State = request.form("State")
        Submitteremail = request.form("SubmitterEmail")
        Comments = request.form("Comments")

        'problemet har nog att göra med recset
        Set RecSet = Server.CreateObject("ADODB.Recordset")
        SQLQuery = "INSERT INTO Req(Typ, Title, Description, IssuerEmail, Date,
State, Comments, Father) VALUES('" & Typ & "', '" & reqtitle & "', '" & Description
& "', '" & Submitteremail & "', '" & Date & "', '" & State & "', '" & Comments &
"', '" & Parent & "');"
        set RecSet = connect.execute(SQLQuery)

        SQLQuery2 = "SELECT MAX(idnbr) AS reqNbr FROM req"
        RecSet.Open SQLQuery2, Connect, adOpenStatic, adLockOptimistic
        requirementNbr = RecSet("reqNbr")
        RecSet.Close

        SQLQuery3 = "INSERT INTO list(project, req) VALUES('" & projectNbr & "', '" &
requirementNbr & "');"
        set RecSet = connect.execute(SQLQuery3)

    End If
    Connect.close
%>

<H1><font size="4"><b>Requirement Break down Form</b></font></H1>
<HR>
<FORM method="post">
<TABLE cellSpacing=5 width="101%" border=0>
<TBODY>
<TR>
<TD width="20%"><STRONG>Product:</STRONG></TD>
<TD width="36%"><% =projectName %></TD>
<TD width="4%"></TD>

```



```

<HR>
<ul>
  <li><a href="hlreqcollection.asp">Submit a requirement or an idea</a></li>
  <li><a href="search.asp">Search function</a> (Update, Break down or View history
for selected requirement)</li>
</ul>
<ul>
  <li><a href="default.asp">Requirement Service start point</a></li>
</ul>
<p><font size="1">Prototype produced by <a href="mailto:cdictionary4@cpen.com">
Urban Martinsson and Åsa
Karlsson</a> as part of our Bachelor Thesis<br>
Last updated <!--webbot bot="Timestamp" s-type="EDITED"
s-format="%Y-%m-%d" --> </font> </p>
</BODY></HTML>

```

showhistory.asp

```

<html>
<!-- #include virtual="adovbs.inc" -->

<% set Connect = server.createobject("ADODB.Connection")
Connect.open "PROVIDER=SQLOLEDB;DATA
SOURCE=192.168.2.192;UID=guran;PASSWORD=urban;DATABASE=RHandler"
Set RecSet = Server.CreateObject("ADODB.Recordset")

ReqNbr = Request.QueryString("requirement")

Requirement = "SELECT ReqHistory.* FROM ReqHistory WHERE ReqHistory.reqId = '" &
ReqNbr & "'"

RecSet.Open Requirement, Connect, adOpenStatic, adLockOptimistic %>

<h1>
<font size="4"><b>History list for selected requirement
</b></font>
</h1>
<hr width="1648">
<table border="1" width="1648">
  <tr>
    <td width="35" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Id</b></td>
    <td width="163" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Title</b></td>
    <td width="145" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Description</b></td>
    <td width="90" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Type</b></td>
    <td width="80" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>State</b></td>
    <td width="61" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Cost
estimate</b></td>
    <td width="78" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Technique
level</b></td>
    <td width="73" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Salability</b></td>
    <td width="76" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Customer
need</b></td>
    <td width="57" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Priority</b></td>
    <td width="52" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Parent</b></td>
    <td width="152" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Additional comments</b></td>
    <td width="108" bordercolor="#FFFFFF" valign="bottom"
align="left"><b>Date</b></td>

```

```

        <td width="194" bordercolor="#FFFFFF" valign="bottom" align="left"><b>Issuer
Email</b></td>
    </tr>
</table>

<% Do Until RecSet.EOF %>

<table border="1" width="1648">
    <tr>
        <td width="35" bordercolor="#FFFFFF" align="left"><% =RecSet("reqId") %></td>
        <td width="163" bordercolor="#FFFFFF" align="left"><% =RecSet("Title") %></td>
        <td width="146" bordercolor="#FFFFFF" align="left"><% =RecSet("Description")
%></td>
        <td width="90" bordercolor="#FFFFFF" align="left"><% =RecSet("Typ") %></td>
        <td width="80" bordercolor="#FFFFFF" align="left"><% =RecSet("State") %></td>
        <td width="61" bordercolor="#FFFFFF" align="left"><% =RecSet("RCost") %></td>
        <td width="78" bordercolor="#FFFFFF" align="left"><% =RecSet("RTechniquelevel")
%></td>
        <td width="73" bordercolor="#FFFFFF" align="left"><% =RecSet("RSalability")
%></td>
        <td width="76" bordercolor="#FFFFFF" align="left"><% =RecSet("RCustomerneed")
%></td>
        <td width="57" bordercolor="#FFFFFF" align="left"><% =RecSet("priority")
%></td>
        <td width="52" bordercolor="#FFFFFF" align="left"><% =RecSet("Father") %></td>
        <td width="152" bordercolor="#FFFFFF" align="left"><% =RecSet("comments")
%></td>
        <td width="108" bordercolor="#FFFFFF" align="left"><% =RecSet("Date") %></td>
        <td width="194" bordercolor="#FFFFFF" align="left"><a href="mailto:<%
=RecSet("IssuerEmail") %>"><% =RecSet("IssuerEmail") %></td>
    </tr>
</table>

<% RecSet.MoveNext
    Loop
RecSet.Close

Set RecSet = Server.CreateObject("ADODB.Recordset")
ReqNbr = Request.QueryString("requirement")
Requirement = "SELECT Req.* FROM Req WHERE Req.IdNbr = '" & ReqNbr & "'"
RecSet.Open Requirement, Connect, adOpenStatic, adLockOptimistic
%>

<table border="1" width="1648" bordercolor="#C0C0C0" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0">
    <tr>
        <td width="35" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("IdNbr") %></td>
        <td width="163" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("Title") %></td>
        <td width="146" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("Description")
%></td>
        <td width="90" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("Typ") %></td>
        <td width="80" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("State") %></td>
        <td width="61" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("RCost") %></td>
        <td width="78" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("RTechniquelevel")
%></td>
        <td width="73" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("RSaleability")
%></td>
        <td width="76" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("RCustomerneed")
%></td>

```

```

        <td width="57" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("priority") %></td>
        <td width="52" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("Father") %></td>
        <td width="152" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("comments") %></td>
        <td width="108" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><% =RecSet("Date") %></td>
        <td width="194" bordercolor="#C0C0C0" align="left" bgcolor="#C0C0C0"
bordercolorlight="#C0C0C0" bordercolordark="#C0C0C0"><a href="mailto:<%
=RecSet("IssuerEmail") %>"><% =RecSet("IssuerEmail") %></td>
    </tr>
</table>

<%
    RecSet.Close
    Connect.Close
%>

<p>
    The shaded area shows the actual status for the selected requirement
<hr width="1648">
<ul>
    <li><a href="listreq.asp">Chose another requirement</a>
    </li>
</ul>
<ul>
    <li><a href="hlreqcollection.asp">Submit a requirement or an idea</a></li>
    <li><a href="search.asp">Search function</a> (Update, Break down or View history
for selected requirement)</li>
</ul>
<ul>
    <li><a href="default.asp">Requirement Service start point</a></li>
</ul>
<p><font size="1">Prototype produced by <a href="mailto:cdictionary4@cpen.com">
Urban Martinsson and Åsa
Karlsson</a> as part of our Bachelor Thesis<br>
Last updated <!--webbot bot="Timestamp" s-type="EDITED"
s-format="%Y-%m-%d" --> </font></p>

</html>

```