How to Diagnose a Control System Remotely

Investigation using Requirements Engineering and Evaluation Techniques

Cumhur Sel

Master Thesis
Stockholm, Sweden 2011

XR-EE-ICS 2011:016
**Abstract**

This Master Thesis has investigated how one should define requirements for a desired product and evaluate the possible solutions, in order to deliver sufficient information to the organization to use as basis. The company’s problem is that there is no communication link to use, between the delivered control systems and the company, in order to support it. The company desires a new product that could be implemented in the control systems worldwide for communication with the control system and for maintenance reasons. The Thesis has therefore started by eliciting requirements for the desired product in order to specify a Requirements Specification. The Requirements Specification has been used to form an evaluation process, and the tailored evaluation process has in turn lead to findings for the possible solutions. The intention with this report is to use this document as a sufficient reference to determine if resources should be committed to constructing or purchasing the product and using it within their organizational context.
# Table of Contents

1 Introduction ........................................................................................................... 5  
  1.1 Background......................................................................................................... 5  
  1.2 Purposes and goals ............................................................................................. 5  
  1.3 Delimitations ...................................................................................................... 6  

2 Method .................................................................................................................. 7  
  2.1 Method overview ................................................................................................. 7  
  2.2 Method weaknesses and possible alternative methods ...................................... 7  

3 Theory and literature review ................................................................................ 12  
  3.1 Requirements Engineering processes ................................................................ 12  
  3.2 Techniques and methods for Requirements Engineering processes ............... 15  
    3.2.1 Elicitation techniques and methods ................................................................. 15  
    3.2.2 Analysis techniques and methods ................................................................. 18  
    3.2.3 Specification techniques ............................................................................... 22  
    3.2.4 Validation methods ....................................................................................... 25  
    3.2.5 Management techniques ............................................................................... 25  
  3.3 Evaluation process ............................................................................................ 26  
    3.3.1 Effective strategies ......................................................................................... 27  
    3.3.2 The PECA process ......................................................................................... 27  
  3.4 Prototype testing ............................................................................................... 30  

4 Development of Requirements ............................................................................ 31  
  4.1 Requirements Elicitation .................................................................................. 31  
  4.2 Requirements Analysis ..................................................................................... 32  
  4.3 Requirements Specification .............................................................................. 35  
    4.3.1 Communication ............................................................................................... 35  
    4.3.2 Information ..................................................................................................... 37  
    4.3.3 Maintenance work ......................................................................................... 37  
  4.4 Requirements Validation .................................................................................. 38  
  4.5 Requirements Management ............................................................................. 38  

5 Evaluation and results.......................................................................................... 39  

6 Analysis ................................................................................................................ 43  
  6.1 Requirements Engineering ............................................................................... 43  
  6.2 Evaluation .......................................................................................................... 44  
    6.2.1 Evaluation of P1, Remote Access Platform .................................................. 45  
    6.2.2 Evaluation of P2, Remote Service .............................................................. 45  
  6.3 Prototype ........................................................................................................... 46  

7 Discussion ............................................................................................................ 47  

8 Conclusions .......................................................................................................... 51  

9 Abbreviations ....................................................................................................... 52  

10 Appendices ......................................................................................................... 53  
  10.1 Appendix A, Process Map – Trip Management .............................................. 53  
  10.2 Appendix B, Process Map – Upgrade due to Manufacturing Fault .......... 53  
  10.3 Appendix C, Process Map – Change Control Board .................................. 53
10.4 Appendix D, Context Diagram – Remote information collection ....................... 53
10.5 Appendix E, Context Diagram – Remote upgrade ............................................ 53
11 Bibliography ........................................................................................................ 54
1 Introduction
In this first section of the final report, the Master Thesis is introduced and motivated. The main purposes and goals are defined and the delimitations are described.

1.1 Background
The master thesis is performed on a business unit at a company that is one of the world’s leading providers of power transmission systems. They deliver technical solutions that directly increase safety, capacity and flexibility of electric power transmission systems.

The business unit delivers turn-key projects worldwide for its customers in form of Static Var Compensators (SVC) and Series Compensators (SC). To be able to support these delivered projects one need connection to these sites in order to retrieve information for further investigation.

As we all know we have demands and needs that need to be fulfilled in order for us to buy or invest in a product. These demands and needs are usually transferred into requirements, mostly in development and procurement, so that they can describe necessary and sufficient properties of that specific product we want. Therefore, it is important to be able to write good requirements so every involved party understands these and this is usually called Requirements Engineering (RE). Otherwise the probability is high to get something we don’t want. Requirements that is missing, incorrect, incomplete or conflicting results often in: (Gottesdiener, 2005) (Young, 2004)

- Dissatisfied customer
- Cost overruns
- Expensive rework
- Late delivery
- Exhausted and demoralized team members, and
- Poor quality

One can eliminate this kind of things by trying to define the requirements in a way so that the receiver understands them properly.

1.2 Purposes and goals
The situation today is that the business unit has no direct communication link with customer sites. Use of a communication link could be to retrieve or collect information for a situation like triggering fault or alarm on customer site etc. If this happens, the customer itself gets a notice of triggering faults in order to solve them by themselves or by getting in contact with a support engineer at the business unit. To be able to get assistant from a support engineer there are log files that needs to be sent from corresponding customer site to the business unit before any actions can be taken. For an upgrade situation the upgrade file needs to be sent to customer or usually a support engineer needs to travel to upgrade on site. The current focus is to streamline the customer support so a support engineer or a specialist can diagnose the customer site remotely. This has led to a proposal of investing in a new function that can take care of it.
The goal of the Master Thesis is:

- to elicit requirements in order to write a Requirements Specification,
- evaluate possible solutions, and
- if a solution matches, a prototype is to be set up at the laboratory to start the testing process.

Another intention with this Master Thesis is to try to design and develop a framework for RE that is general enough, so it can be used in similar projects. RE is a common topic in projects and there are research covering the area, which will be discussed in chapter 3.

1.3 Delimitations

The project has a time budget and it must be kept. RE concept have many techniques and methods that one can select from and use in projects. In order to maintain the time budget the focus has instead been on choosing and carrying out few relevant ones for each process throughout the project.

As there is no single way of carrying out RE, it is possible that there could be important factors that have not been observed. When making choices on what to include and what to exclude in this project, the solution for the business unit have been in consideration.

While it to some degree may have been possible to draw general conclusions from some results of this project, this report has been written with the solution in mind. Having stated that, it is worth noting that the RE framework used in this project is designed to be applicable to similar projects with as few modifications as possible.

Section 1.2 describes that there is no communication link between customer site and the business unit. This communication link could be in form of a modem, VPN, or satellite connection etc. However in this project the communication link has been considered when eliciting and specifying requirements but has not been tested in the prototype, communication was instead assumed to be in place.
2 Method
In this section, the methods that are used in the project are presented. There is a discussion on the strengths and weaknesses of the RE tools chosen for this project and possible alternative methods. The used methods for the evaluation are also discussed here.

2.1 Method overview
This project started initially by reading literature related to the RE and documents about the control system. The literature consisted mainly of books and articles related to RE and to some extent of evaluation, but also some documents to get an insight in how the business works at the company. The literature has been read and considered from a critical point of view.

The framework for the project presented in this report is based on the RE framework developed by Gottesdiener (introduced in section 3.1), the framework is however modified to some extent to adapt to this kind of projects.

The first main part of the project was to develop a Requirements specification. The second part was to evaluate two possible solutions compared to the developed Requirements specification, which in turn should have led to a prototype testing in the end.

The two possible solutions were however kept in secret until the Requirements specification was finished so the Requirements specification wouldn’t be biased from the impressions of any solutions. The project has therefore been logarithmic planned, i.e. the project was planned in greater detail for the RE part than the evaluation and prototype testing part in the beginning, and the remaining parts were planned when the RE part was about to end.

2.2 Method weaknesses and possible alternative methods
The choice to use Gottesdiener’s framework as basis was based on earlier study and knowledge, and the framework can be adapted to different kinds of projects.

There are several methods and techniques for RE so one can modify the framework in their own way. Gottesdiener discusses and describes RE in form of five processes elicitation, analysis, specification, validation and management of requirements. These processes have been the basis and it is the execution that has been adapted to this project.

According to Gottesdiener elicitation process identifies the sources to elicit requirements from, but also evokes the requirements from these sources. Table 1 shows different techniques for how an elicitation can be performed, and how much these techniques cost for involved parties and if they give tacit knowledge for the requirements engineer. The project scope and the table should be in mind when deciding which elicitation techniques to use. Having that in mind following techniques have been chosen for this project

- **Interviews**: good way to begin the elicitation with, by choosing few people that can give initial information on requirements on a higher level but also from whom to elicit requirements from,
- **Observation**: shows how the work is performed today, and can give information on how it could be improved with help of the new function, and
Existing document studies: documented business processes related to the project are of aid when eliciting requirements and it gives an insight on how the business works.

Elicitation techniques are discussed and described in greater detail in section 3.2.1 of this report.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Cost for requirements engineer</th>
<th>Cost for company</th>
<th>Tacit knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interviews</td>
<td>Medium</td>
<td>Medium</td>
<td>Yes-No</td>
</tr>
<tr>
<td>Exploratory prototypes</td>
<td>Medium-High</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Facilitated workshops</td>
<td>Low-Medium</td>
<td>Medium</td>
<td>Yes-No</td>
</tr>
<tr>
<td>Focus groups</td>
<td>Medium</td>
<td>Medium</td>
<td>Yes-No</td>
</tr>
<tr>
<td>User task analysis</td>
<td>High</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Observation</td>
<td>High</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Existing document study</td>
<td>High</td>
<td>Low</td>
<td>No</td>
</tr>
<tr>
<td>Surveys</td>
<td>Low</td>
<td>High</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1, Summary of costs for elicitation techniques (Kotonya, 1998)

To effectively analyse requirements the requirements engineer need to understand and define the elicited requirements in a form so that stakeholders can prioritize them based on their needs. The analysis will result in models, tables and diagrams, which will be used to represent the user requirements. To provide richer insight into requirements one should choose models that answer multiple focus questions, i.e. What, How, When, Why and Who. (Gottesdiener, 2005)

- **Process Map** is used to model the business to get a better insight into what inputs, outputs and sequence steps there is across multiple functions, organization, or job roles.
- **Context Diagram** helps understand the project scope. The diagram represents the system in its environment with external entities, i.e. people and systems that will provide and receive information to and from the system.
- **Prioritization of requirements** is performed to rank one requirement over another. The importance of this step is to negotiate trade-offs among requirements between stakeholders. Prioritization emphasizes the requirements that are of higher importance for the function, which in turn will increase the business value.

Prioritized requirements can then be specified in a document, so-called Requirements Specification. The specification is a process where requirements are refined, elaborated, and organized for its intended readers, e.g. users, developers, head offs etc. Adapting the specification to its readers is difficult and some writing techniques have been considered when specifying requirements. The approach of how to specify the requirements is described and discussed in section 3.2.3.
Validating the requirements certifies an acceptable description of the function that is to be implemented. Validation can be performed by different techniques and one that is used often in this project is peer reviewing. Peer reviewing is a focused meeting where evaluation of developed models and specification is performed. They are often performed by a small group of stakeholders, but could for some cases be performed together with interviewed interviewees, if nothing else are decided.

Another technique, operational prototype, can be used to validate the requirements. The prototype can also be used to evaluate the possible solutions but is often performed both in the validation and the evaluation phase. In the evaluation phase the prototype is often used to evaluate the solution for its major features and functionalities, see section 3.3, before taking decision on “Go!” or “No go!” for implementation of the new function. The advantages are that prototypes present the true system and one can’t neglect the problems with it, the disadvantage however is that prototypes are usually very expensive.

Parallel to the RE processes management of requirements is performed. The management is a process where requirements are monitored and controlled for status and changes. This parallel on-going phase is very important to keep track since requirements are not stable and fixed. Requirements are instead very often volatile and require procedures to quickly understand impact of changes. If a change occurs it is very important to trace these changes. Management handling is mainly performed by the requirements engineer but included also the supervisors for this project. Management handling often requires a Change Control Board (CCB) and in this project the board members consisted of Mr. Sel together with the supervisors. The process steps for change control procedures are illustrated in Figure 1.

![Figure 1, Process map for requirements change control procedures (Rocha Flores, 2011)](image)

Section 3.2.4 will give an introduction to how one can validate requirements. As we know there are two solutions the company have been looking at. These solutions were as described before not presented until the Requirements Specification was approved. The evaluation and prototype processes could
therefore not be planned and described in greater detail for the project until the solutions were presented.

The first evaluation process consisted of finding matches between the given solutions and the specified requirements. If one solution were to match the specified requirements, a prototype setup and testing would begin for further investigation and evaluation. An evaluation of the prototype is often performed to check and control how well the product matches the specified requirements. It is also used for some requirements that can’t be evaluated and compared beforehand and needs to be checked in its working environment.

The evaluation process was based on evaluation strategies and the PECA process developed by the Software Engineering Institute in Pittsburgh, US (Comella-Dorda, 2004), described in section 3.3, but had to be adapted for this project’s purpose and time. The PECA process consists of four evaluation activities:

1. Planning
2. Establishing criteria
3. Collecting data
4. Analysing data

The evaluation of the two possible solutions was planned to be simple and be easy to perform in order to keep up with the time plan for the project and evaluate the two solutions rapidly. In larger evaluations there is often an evaluation team selected with different stakeholders to create a neutral evaluation as possible. In this project the evaluation was performed by Mr. Sel and reviewed together with the supervisors for this project. The approach for the evaluation could not really be determined as in a common evaluation because there were already two possible solutions to be evaluated. However it could be determined to use the “Best-fit” approach instead of “First-fit”, i.e. select the solution that has a best match instead of selecting the first solution that matches the specified requirements.

The depth of the evaluation is also an important factor to consider and in this case it was decided to take use of the prioritization for the requirements. The requirements are prioritized from most important features, grade A, to least important, grade D. To be able to determine the depth of the evaluation the requirements that were most important was evaluated first, i.e. if one solution should weigh up after evaluating the A-graded requirements the evaluation could be considered as complete. If not, it would continue to the requirements with the next grading level, i.e. B. This procedure should go on until one solution weigh over the other and if this should end up in a tie; the evaluation would have to be continued in evaluating the prototypes for the two solutions. Evaluation of a prototype is of aid when there are requirements that can’t be evaluated directly by reviewing specifications and documents.

Another possible scenario is if none of the two solutions matches the requirements, then a market research would be performed.
The second activity in the PECA process is establishment of criteria. This activity is divided into three steps according to PECA; defining evaluation requirements, defining evaluation criteria and prioritizing criteria. For this project, the process has been different, the requirements have been used as evaluation criteria and the prioritization from the requirements specification has been used. Section 3.3.2.2 describes how one could use the requirements as criteria and which aspects to consider.

Once the criteria are defined one can start collecting data for the evaluation. Data can e.g. be collected from literature reviews, hands-on techniques and benchmarking, please see section 3.3.2.3 for further descriptions. Literature reviews have been the main source for data collection, but there was also a prototype testing planned. The literature consisted of presentation documents and product specifications, the presentation documents have been reviewed critically since they often seem to omit the weak parts of the product and enhance the strong parts. Another data collection technique, prototype testing, were planned but could not be performed due to difficulties in establishing contact with the vendor, which in turn lead to lack of time. The intention with the prototype testing and implementation was to evaluate the product in its working environment, i.e. how the product would interact with other system components, so that the major functionalities could be tested and evaluated in action.

The collected data for the evaluation does not include raw data. The evaluation has instead been based on finding matches between the defined criteria (requirements) and the two solutions (“Best-fit”). The analysis of the collected data will therefore be in a form of a Gap analysis, which is based on finding gaps between the end-user processes and the processes assumed by the products. The results from this type of an analysis is best presented in a matrix, please see section 6.2.
3 Theory and literature review

The definition of a requirement is defined as

“A requirement is a necessary attribute in a system. It is also a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have value and utility to a user.” (Young, 2004)

RE is used in procurement and development projects. Typically projects where requirements are used could be maintenance of a system, development of a new system and procurement of an existing software (so called Commercial-off-the-shelf) etc. The main idea behind RE is to keep the upcoming expenses relatively low. Figure 2 represents the cost for detecting and fixing errors in the different phases of a procurement or development project.

This chapter will give an introduction to RE and describe the phases, techniques and methods that come with it and how they should be implemented.

3.1 Requirements Engineering processes

(Gottesdiener, 2005) is written in order to provide a resource that one involved in RE can use to communicate needs and share understanding with each other. The book can be used as a quick reference guide regarding RE to explore practical steps, examples and tips. All stakeholders at any organization level, e.g. analysts, project managers, software developers, and business managers, who are involved in projects where RE is a one of the subjects, can use it (Gottesdiener, 2005). Together with related literature, such as (Young, 2004) (Kotonya, 1998), and (Alexander, 2002) etc. have been the basis for the RE in this project. Further on, the processes are introduced and described.

Elicitation of requirements can be performed with several techniques and methods (please see section 3.2.1). By using several methods and techniques one is able to ensure high reliability and validity. The first thing to do is to select and plan the techniques and methods that will be used. The second step is to set goals and expectations for chosen methods and prepare them. The next step will be to begin elicitation process and document findings carefully. The final step is to verify and correct acquisitions, and if any inconsistencies are to be found one should perform another elicitation for relevant requirements. If consistencies are cleared and one is able to complete final step for elicitation process, it is time to proceed to the analysis phase. Illustration of the elicitation steps can be found in Figure 3. (Gottesdiener, 2005) (Kotonya, 1998)
The next phase will help us to analyse our elicited requirements. With help of some techniques, such as different maps, tables and diagrams etc., we will be able to analyse requirements. There are several ways to analyse requirements, as mentioned and only few of them will be used, please see section 3.2.2 for further information.

As we can see in Figure 4, analysis starts with modelling the business, to give an insight on how the business works. Following step is to defining the scope for the project and proceeds to creation of detailed user requirements models these steps shall be carried out with aid of different models and maps etc. The final step for analysis will be, as the name implies, Prioritization of the requirements. By prioritizing requirements one is able to grade requirements based on their value for the business. The prioritization is a very well discussed subject by other authors, cf. section 3.2.2.6 for more information. (Gottesdiener, 2005) (Kotonya, 1998)

As one can see from Figure 4, one needs to iterate the previous step if any inconsistencies or ambiguities are detected. To verify and correct these findings stakeholders can be consulted again.

Specification of requirements consists of two parts, User requirements and Software requirements, that each specify what the user requirements and the software requirements for the system are. By dividing them into two separately parts one can assure a final check of requirements and specify them in an accessible way for its developers and future users. (Gottesdiener, 2005)
This will describe, from a user point of view, how it is intended that the users will experience the requirements specified when they use the system. This is followed by a verification of user needs that coincides with the intended purpose of the system. Direct users will be used to verify the quality of the document in order to make the verification more efficient. The verification will be used to detect flaws in the document, in order to take actions on changing the specification of requirement, to make it more accurate. 

Documentation of software requirements describes requirements, from a developer’s point of view, in a way that can be used by the system developers to create the system as well as being used as a specification when signing a contract with the vendor. Parts of the IEEE 830-1998 standard may be used as a template for the Software requirements specification for the system. The last step is to verify software requirements, i.e. check so that requirements are correctly specified, formulations are properly made, but also so that it can provide a sufficient base to begin the development of the system. (Gottesdiener, 2005)

If flaws regarding previous requirements processes or steps are uncovered, flaws must be corrected and affected document must be modified before continuing on any further step or process. Some flaws can also result in redoing steps in previous processes.

The next phase will help to verify whether the product satisfies user needs and corresponds to the specified requirements, i.e. detecting and correcting missing, incorrect and/or unnecessary requirements. Another essential factor that validation covers is that necessary and sufficient requirements are specified, in order to meet stakeholder’s needs before development commences. (Gottesdiener, 2005) (Kotonya, 1998)

As illustrated in Figure 6, validation consists of few steps, and can be accomplished with help of methods and techniques, please see section 3.2.4 for further information.

![Figure 6, Adaptation of requirements validation (Gottesdiener, 2005)](image)

This phase begins with selection and integration of requirements validation techniques, i.e. decide which technique will be most effective, time reducing and which high-risk requirements will be validated (high-risk requirements tend to have huge impact on projects and the risks that derives can be reduced by handling them first). The next will be to ensure adequate user involvement, i.e. check if derived requirements are functional, check with the stakeholders that defined requirements are unambiguous, consistent, complete etc. and that requirements for how users’ needs to interact with the system is accurately described. Step three consists of validation of requirements, this means that requirements engineers shouldn’t wait until detailed analysis models are completed, instead they should start validation of important requirements in an early phase so that right description of product is ensured. The last step is to revise the requirements documentation, i.e. as soon as feedback is received one should immediately start revising the document. The
feedback can also contain changes, one must then carry out impact analysis before changes are implemented and reprioritize them. This step should be repeated as one progress through development process. (Gottesdiener, 2005)

Management of requirements is however an on-going project phase during Elicitation to Validation of requirements. The phase includes several things that could be implemented to develop a Requirements Specification document. One major issue on all types of projects is to control and follow up changes of requirements. Another one is to understand requirements lineage and relationships. These types of questions can be answered with different techniques and methods, e.g. Change control boards and Requirements matrices, please see section 3.2.5 for further information. (Gottesdiener, 2005) (Kotonya, 1998)

3.2 Techniques and methods for Requirements Engineering processes

As discussed in previous section, Gottesdiener, Kotonya & Sommerville and Young describes that there exists several different kinds of techniques and methods to carry out the RE processes with. This section will try to give an insight and describe these techniques and methods to help the reader understand the underlying literature behind the chosen method.

3.2.1 Elicitation techniques and methods

In this section the different elicitation techniques are presented and described in greater detail.

Table 2 summarizes the elicitation techniques from the different sources.

<table>
<thead>
<tr>
<th>Sources</th>
<th>Gottesdiener</th>
<th>Kotonya &amp; Sommerville</th>
<th>Young</th>
<th>Bray</th>
<th>Maciaszek</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholder categorization and profiling</td>
<td>✯</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interviews</td>
<td>✯</td>
<td></td>
<td>✯</td>
<td>✯</td>
<td></td>
</tr>
<tr>
<td>Surveys</td>
<td>✯</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observation</td>
<td>✯</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Existing document studies</td>
<td>✯</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2, Summary of techniques from sources

3.2.1.1 Stakeholder Categorization and Profiling

To be able to elicit requirements one has to identify sources that requirements can be elicited from. Categorization and profiling of stakeholders is a common way of identifying and listing the relevant sources so one can get an overview. (Gottesdiener, 2005)

3.2.1.2 Interviews with stakeholders

One very common way for elicitation of requirements is interviewing stakeholders. It is also a broad method to identify requirement topics such as user
environment, business goals etc. but also to identify additional sources of requirements. (Gottesdiener, 2005) (Kotonya, 1998) (Alexander, 2002) (Young, 2004) (Bray, 2002)

Before starting preparing questions one has to identify interviewees that could contribute to elicitation of requirement topics of a broad perspective, i.e. the interviewer should try to use stakeholders from different types of backgrounds (subject matter experts, sponsors, users etc.). Suitable stakeholders for the interviews could be: (Gottesdiener, 2005) (Kotonya, 1998) (Alexander, 2002) (Young, 2004) (Bray, 2002)

- Chief Operating Officer, COO
- Heads of departments or business units
- One or two workers from each department
- Contractors

One can now start prepare the questions, but have in mind that interviews could be of different kinds, e.g. in-depth interviews, were the questions are quite detailed or a focus interview that has, as the name implies, a clear focus and should be brief, and the questions although maybe still open-ended, should be based almost entirely on the purpose of the interview and the interviewee. (Ekholm, 1975) (Dahlström, 1957) Both should be used for different kind of interviewees. When this preparation is completed, interviewees needs to be scheduled, this step should be done iteratively for each interviewee so that one can design relevant upcoming questions for the next interview session or interviewee. (Gottesdiener, 2005) (Yin, 2009) (Kotonya, 1998) (Alexander, 2002)

There are a couple of things to consider when performing interviews so answers from the interviewee won’t be biased. That is thing like taking notes, listen actively, avoiding leading questions (Dahlström, 1957 pp. 129-136), start with general questions which are gradually followed by more specific ones (Ekholm, 1975 pp. 60-62), try to have a neutral relationship to the interviewee (Dahlström, 1957 p. 130), etc.

Final step will be to document all results and it should be done immediately after the interview to be able to recall all the answers and discussions that have been made. If there are any ambiguities these should be resolved with the interviewee. Notes that are handwritten have to be digitalized so it becomes easier to store and backup the documents. (Gottesdiener, 2005) (Yin, 2009)

The activities to elicit requirements from interviews with stakeholders can be seen in Figure 7 below.

Figure 7, Interview phases (Gottesdiener, 2005)

3.2.1.3 Surveys

Surveys are a great complement to interviews, because one reaches out to a bigger segment and it isn’t time-consuming as interviews for the investigator but is costly for the company (Gottesdiener, 2005) (Yin, 2009). However, it is a good way to accomplish data triangulation, because when results of surveys have been conducted and compared to the interview results, it has been found that people
tend to be more honest when responding to surveys than interview questions, especially when you perform anonymous surveys. (Dahlström, 1957 pp. 137-139)

Before elaborating a survey one must have a purpose. One of the projects purpose is to specify a requirements specification but a surveys purpose must be divided into more discrete goals so one is able to collect detailed information, e.g. obtaining feedback on proposed features etc. When this is accomplished the right sample group or groups must be chosen to get as much as possible from the survey. (Gottesdiener, 2005)

Surveys can be distributed in several ways, one way is to hand out the survey for the focused group, another is to email them or you could just put up a web-form. The task before distributing surveys is to design and test it, focus must be on the sample group. Questions design could be of form: multiple choices, users left to put the number or users allowed to writing free-form text. To avoid bias and ambiguousness the survey should be structured and designed with different versions for the same questions and not using leading questions, short questions should be used instead. To ensure validity, reliability and understandability, but also how long the survey will take, few real respondents should test it. This is to get feedback and revise before distributing the survey. (Gottesdiener, 2005) (Yin, 2009)

Surveys should include a cover letter or some kind of introduction before the survey starts so that the respondent gets an insight of what the survey is for, how the results will be used, how long it could take and contact information. Another factor here is to inform the respondent if the survey is anonymously or not, since anonymity has a huge impact of the answers. (Dahlström, 1957)

To be able to use the results one must have an insight in reliability and repeatability and of course the statistic outcome of the results. These could be considered with external help or a study group consisting of survey designers and project members (maybe also testing respondents). (Gottesdiener, 2005)

3.2.1.4 Observations

Observations are when one observes the potential users of the becoming solution in their current everyday work (Yin, 2009). This is performed in order to get clarification how the users will become to use the solution when it is implemented, and also as verification of previously gathered requirements. (Gottesdiener, 2005) (Maciaszek, 2007 p. 84)

There are three types of observations according to (Maciaszek, 2007) and (Yin, 2009):

- **Active observations**: the requirements engineers actively take part in the everyday work
- **Passive observations**: the requirements engineers simply observe without interaction with the user
- **Exploratory observations**: the user explains the work while performing it
One thing to take into consideration during an observation is that users might perform their work differently from how they normally do. Often the observer doesn’t have the necessary knowledge to perform the active observations, in these cases it is highly recommended to combine active, passive and explanatory observations.

Suitable stakeholders to perform observations on are often direct users, direct users are the ones that perform the work and will use the solution actively.

The steps for performing an observation are to begin with identifying suitable users. Second is to schedule and prepare the observation, such as planning what tasks the users should perform. Third is to conduct the information before moving on to the final step, i.e. analyse and document the observation. One should though have in mind to inform, the user that one is going to observe, when and how the observation will be carried out. (Gottesdiener, 2005)

![Figure 9, Phases to perform observations (Gottesdiener, 2005)](image)

### 3.2.1.5 Existing document studies

Existing documentation study is a task performed by studying the documentation available in the organization in order to discover requirements relevant for the upcoming solution (Yin, 2009). Already existing documentation is adapted to the realities of the company, a study of it will make it easier to find requirements that makes it better integrated with the company organization.

Studying existing documentation is a good way to create a basis for a requirements specification and complement possible shortcomings of other elicitation methods (Gottesdiener, 2005). Complementing documentation studies in the beginning and throughout the project is a good base for the RE processes. Depending on the documentation available and what needs to be complemented, all stakeholders can be consulted in finding suitable documentation.

There is also another possibility to gather requirements from, and that is from existing COTS-systems. Therefore it could be of interest to investigate similar solutions.

### 3.2.2 Analysis techniques and methods

#### 3.2.2.1 Relationship Map

Relationship maps shows what information and products are exchanged among external customers, providers, and key functions in the organization. The map is used to identify affected business functions and their inputs and outputs to get an organizational background for the project.

The processes for making a relationship map are to first identify key functions, departments and work groups, such as Customer Support, Technical department, Providers, Finance etc. and draw them as boxes. Next step is to list each functions or departments core inputs and outputs, i.e. what they receive and produce. Last step is to draw relations between the boxes with the corresponding core input and/or output, the relations should be on a high level, e.g. relation...
between Maintenance team and Finance could be report expenses. (Gottesdiener, 2005)

Relationship maps should be carried out together with few stakeholders from different background in the organization to get as much input as possible.

Figure 10 illustrates an example of how a relationship map could look like for operating a course on a Technological Institute in Sweden. A software tool that can be used is Microsoft Visio.

![Figure 10, Example of Relationship Map (Group I, 2010)](image)

### 3.2.2.2 Process model

A process model shows the sequences to handle a business process across multiple functions, organization, or job roles. These business processes can be described with help of modelling languages, and one that will be of use is the Business Process Modelling Notation (BPMN). BPMN is based on a flowcharting technique and it creates graphical models of business process operations. The model represents a network consisting of graphical objects, which are activities (inputs and outputs) and the flow controls and defines their order of performance. Swim lanes in the models are used to illustrate different functional capabilities or responsibilities. (Gottesdiener, 2005) (White)

This type of modelling provides a notation that becomes readily understandable by all involved persons in the project.

Figure 11 illustrates an example of a process where BPMN is used. (Gottesdiener, 2005) (Pichler, 2006) (White)
3.2.2.3 Context Diagram

With help of a context diagram one can define the project scope and concentrate on what inputs the system needs to provide outputs. Another main use for the project team is to deduce different requirements models, e.g. Use cases and Dialog maps. Context diagram illustrates how external entities interact with the system in its environment. The core thing is to show how these external entities provide and receive information to and from the system. (Gottesdiener, 2005)

Context diagrams can be performed with help of the software Microsoft Visio and an example is presented in Figure 12 (the example is again an illustration of a Technological Institute in Sweden).

As for Relationship map one can make the diagram together with few stakeholders from different organizational backgrounds. The diagram is supposed to be verified against other models after completion to discover any inconsistencies. This can help to discover and reduce the amount of unnecessary requirements.
### 3.2.2.4 Use Cases

Use cases are step-by-step descriptions that show how users are supposed to use the solution in order to achieve an objective. Use cases demonstrate how users will utilize future functionality of the solution. It is important to include error handling and variations. (Gottesdiener, 2005) (Kotonya, 1998) (Alexander, 2002)

Use cases have the ability to ensure that user requirements are correctly understood, by verifying it with its intended users, and exemplify for developers how the system should work. (Widrig, 2003 pp. 147-164)

The different stakeholder types involved during the use case part of the requirements analysis are commonly direct users, they are the ones who will be using the system and are best suitable for describing and exemplifying performed work. First the stakeholders that will review the use cases will be chosen. Then use cases are created from requirements that have already been gathered. After that each use case is shown to the chosen stakeholders that will come to utilize the functionality in the use case. Finally the feedbacks from the stakeholders are analysed and suitable modifications are performed on the use cases. (Gottesdiener, 2005)

Both text and UML models are suitable for describing use cases, and computer software like Microsoft Visio could be used for digitalization. (Gottesdiener, 2005) (Sommerville, 2007 pp. 124-126)

### 3.2.2.5 Dialog Map

Dialog map is a method similar to use cases but uses a prototype of the systems interface and a map showing transitions between system dialogs, instead of step-by-step descriptions. (Gottesdiener, 2005) (Brennan, 2009 p. 197)

First a complicated use case with many uncertainties or several use cases that have uncertainties regarding how the transition between them should work is picked out. Then an interface prototype is created based on the one or several use cases picked out and how the transitions could look like is determined. Followed by a diagram showing the transitions between different states are created. Verification of the dialog map is the final step, this ensures that all the parts in the map are described in use cases and that the map is consistent with the other used models. (Gottesdiener, 2005)

### 3.2.2.6 Prioritization of Requirements

Prioritization is performed to decide which requirements are more important than others and which requirements depend on other requirements. This ensures that the most important requirements are given the sufficient amount of time and resource, so that all crucial requirements are implemented, when deciding in what order requirements need or should be implemented. (Gottesdiener, 2005) (Brennan, 2009 pp. 99-103) (Kotonya, 1998)

To begin with the requirements that need to be prioritized are picked out and organized. This includes determining which requirements depend on other
requirements, which requirements depend on each other, which requirements that has no dependency connected to them, i.e. are stand-alone, and ensuring that all requirements have the same level of detail. Second is to choose which stakeholders should take part in the prioritization. Suitable stakeholders could be future users, project managers, technical experts etc. When stakeholders has been chosen is it time for the prioritization, but one should first determine criteria for making the decisions about the relative importance of requirements, e.g. customer value, cost of implementation, difficulty of implementation, technical risk, ease of deployment etc. (Gottesdiener, 2005)

![Figure 13, Prioritization of requirements (Gottesdiener, 2005)](image)

The prioritization can be performed with different techniques, below are some of them listed and explained: (Gottesdiener, 2005) (Brennan, 2009 pp. 99-103)

**MoSCoW**: simple prioritization technique which is useful for small projects with few requirements, and the requirements can only be prioritized as Must, Should, Could and Won’t. (Gottesdiener, 2005) (Brennan, 2009 p. 102) (Bittner, 2003 pp. 74-76)

**$100 – bill**: each member of the prioritization meeting is given fictive $100 and is allowed to spend the money on the requirements. The most spent on requirement is the most important. (Rocha Flores, 2011)

**Grading**: assigning each requirement a grade, for instance on a scale 1-5, A - E or high-medium-low. Remember to use few grading levels, define them rigorously and make sure to write the definitions down. (Gottesdiener, 2005)

**Cost-Value approach**: easy to use method for prioritizing software product requirements. The main idea behind is to find a value or a cost comparing one requirement of a pair with the other, e.g. if (Req1, Req2) has the value 2, (i.e. requirement 1 is valued two times more than requirement 2), then (Req2, Req1) gets the value 1/2. (Karlsson, 1997)

### 3.2.3 Specification techniques

The requirements specification is a document used to communicate requirements to the users, project managers, and developers etc. This chapter will describe how one can improve the structure and organisation of the Requirements Specification in order to make the specification more understandable for its readers, but also how the requirements should be specified.

A Requirements specification has a number of sections, and there are several standards one could follow. The IEEE Std.830-1998, (IEEE, 1998), is one standard that can be used in many projects because the standard describes recommended approaches and presents several different outlines for how one could specify the software requirements. Use of the standard can give guidance
for how one should specify and outline the requirements in this project. The IEEE 830-1998 standard recommends that the specification document should include following three main sections:

- Section 1: Introduction of requirements specification documents purposes and scope, but also glossary, acronyms and used abbreviations.
- Section 2: Description of factors that affect the intended solution and its requirements.
- Section 3: Description of the software requirements.

Requirements should therefore be structured as a family of scenarios: each chapter, section and subsection heading is the name of a goal or sub-goal. Each subsection could for example represent one user task that usually contains one functional requirement along with one or two constraints.

Besides the structure writing the requirements is another very important task and there are many things to consider, e.g. each requirement must have a unique identifier, essential requirements must be marked, and requirements must have a priority and a reference so its origin can be traced, and it is also important to have dependencies on requirements. The requirements can be structured in different ways and below is a few examples:

- Partitioning: requirements are structured in “part-of” the function, i.e. divided in what parts the function consists of, e.g. a car consists of a body, engine, wheels etc.
- Abstraction: requirements are structured based on their uniqueness, e.g. if a car is to be specified, the abstraction structure divides the requirements into the different model types, e.g. SUV, Station wagon etc.
- Projection: structures requirements into “views of”, e.g. requirements for a car can be seen from the driver’s perspective and passenger’s perspective, but also from the mechanical or electrical point of view.

When writing requirements one should make sure that each requirement complies with the attributes in Table 3, because studies have shown that, the Requirements Specification readers have understood the context better for requirements that have complied with the attributes than those that have not. (Gottesdiener, 2005) (Kotonya, 1998) (Alexander, 2002)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understandability</td>
<td>Can readers of the document understand what the requirements mean? If the requirement can be understood it can’t be validated</td>
</tr>
<tr>
<td>Ambiguity</td>
<td>Are the requirements expressed using terms, which are clearly defined? Could readers from different backgrounds make different interpretations?</td>
</tr>
<tr>
<td>Completeness</td>
<td>Does the checker know of any missing requirements, or is there any information missing from individual requirements descriptions?</td>
</tr>
<tr>
<td>Consistency</td>
<td>Do the descriptions of different requirements include contradictions? Are there contradictions between individual requirements and overall system requirements?</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Organisation</td>
<td>Is the document structured in a sensible way? Are the descriptions of requirements organised so that related requirements are grouped? Would an alternative structure be easier to understand?</td>
</tr>
<tr>
<td>Redundancy</td>
<td>Is information unnecessarily repeated? Sometimes, of course, repeating information adds to understandability, so balance these.</td>
</tr>
<tr>
<td>Conform to standards</td>
<td>Does the specification and individual requirement conform to defined standards?</td>
</tr>
<tr>
<td>Traceability</td>
<td>Are requirements unambiguously identified, 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?</td>
</tr>
</tbody>
</table>

Table 3, Adaptation of Requirement Attributes (Kotonya, 1998 p. 96)
3.2.4 Validation methods

3.2.4.1 Peer review
Peer reviews are meetings performed by a small group of stakeholders with focus on evaluating requirements and models in order to find errors and improve quality of the requirements specification. Peer reviews

- ensures that team members understanding is consistent with user needs,
- forces validation of the requirements with the highest risk or ambiguity,
- defines quality expectations for requirements,
- improves the understanding of requirements and business domain for stakeholders, and
- identifies requirement process improvement opportunities. (Gottesdiener, 2005)

![Figure 14, Peer Review phases (Gottesdiener, 2005)](image)

3.2.4.2 Acceptance tests
These tests are specific and are often corresponding to use cases developed in the RE processes. User can with help of these tests decide whether to accept the solution or not, i.e. the solution is tested for some inputs that the final solution is designed to execute. The tests are evaluated in statements pass or fail, and the failed tests are ranked in highest priority for correction. Conclusions from the tests help the business to accept or not accept the solution. (Gottesdiener, 2005)

3.2.4.3 Operational prototype
A prototype is built to demonstrate that the solution can satisfy the user needs, i.e. to check in operational form how the solution will work in its future environment. One should consider this type of validation of requirements if time and cost permits, cause prototypes has shown to be a good validation tool but also costly. (Gottesdiener, 2005) (Kotonya, 1998) (Alexander, 2002)

![Figure 15, Prototype phases](image)

3.2.5 Management techniques

3.2.5.1 Change control procedures
Management of requirements in section 3.1 gave us an insight in how important it is to follow and trace changes in requirements. To be able to handle changes
and trace them one has to decide policies and procedures when a change occurs, to establish mechanism and rules for recognizing, evaluating, and deciding how to integrate new and evolving requirements into the requirements baseline. (Gottesdiener, 2005) (Kotonya, 1998)

![Figure 16, Phases for Change Control Procedures](image)

First step is to **identify change control procedures**, this should include things like proposing requirements changes, conducting impact analysis, evaluating trade-offs, making decisions about the proposed requirements, updating documentation and monitoring volatility. But could also be to identify who should be notified of changed requirements for example. In order to follow the procedures for changing requirements one should **create a small group**, usually called **Change Control Board (CCB)**, consisting of technical and business staff to get as different perspectives as possible. Next step is to **create a baseline for the requirements**, i.e. keep some type of record of all information regarding the requirements. Final step is to **begin implementation of the change control processes**, but the most important thing to have in mind is to report and monitor all changes. (Gottesdiener, 2005)

Requirement trace matrix, also known as trace list, is a graphically good way to present tracing of requirements. The matrix identifies the relation between requirements but also how requirements are related to software development deliverables. (Gottesdiener, 2005) (Kotonya, 1998)

### 3.3 Evaluation process

In order to determine which solution match elaborated requirements one has to establish an evaluation process. Available products, functions and systems etc. must be evaluated to determine their suitability for its purpose. Poor and insufficient evaluation results in ending up with unwanted system properties, which often results in costly adaptions and rework and in worst-case cancellation of the entire project.

The process is difficult since there are several perspectives to consider in an evaluation e.g. a system. Evaluations performed by people with different backgrounds will almost always end up with different opinions about the process and its outcome, e.g. an engineer and an accountant, where architecture and design is of great concern for one while cost is for the other one. As one can state it is of greater interest to evaluate the system in context of its intended use rather than just the technical criteria, i.e. an evaluation based on the system in its environment communicating and performing the tasks it’s intended to do.

Table 4 lists common mistakes in an evaluation process. (Comella-Dorda, 2004)
### Table 4, Common Evaluation Mistakes (Comella-Dorda, 2004)

<table>
<thead>
<tr>
<th>Mistake</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inadequate level of effort</td>
<td>E.g. critical systems are selected using only an informal internet search.</td>
</tr>
<tr>
<td>“Once and done”</td>
<td>New product releases or changes in the system context are not reevaluated.</td>
</tr>
<tr>
<td>Non-contextual</td>
<td>“Best of breed” lists or consumer reports are used as the only source of information.</td>
</tr>
<tr>
<td>No hands-on experimentation</td>
<td>Marketing data are accepted as facts without further checking.</td>
</tr>
</tbody>
</table>

#### 3.3.1 Effective strategies

Successful evaluations often consist of the following strategies: (Comella-Dorda, 2004)

- Context consideration: The evaluation should be carried out close to the actual usage, to acquire an accurate evaluation of the system.
- Uncertainty accountant: Minimize unforeseen potential errors with different methods such as risk management and mitigation strategies.
- Comprehensive conduction: Make the evaluation comprehensive, try to use different approaches/views to evaluate the system and do not just use specifications. Have though the available resources in mind when planning the evaluation, exceeding time budget is a risk one always wants to avoid.
- Continuous evaluation: To get better understanding of system that is supposed to be procured, changed or upgraded the evaluation should be continuous, i.e. repeat tasks to improve the understanding of the system in order to minimize the uncertainties.
- Facts guidance: Establish the evaluation based on verified facts, try to neglect opinions and impression to get an accurate evaluation as possible.
- Follow defined evaluation process: Following a good consistent process results often in an accurate evaluation. A good evaluation process should be
  - cost-effective,
  - iterative,
  - include different criteria, and
  - simple

#### 3.3.2 The PECA process

The PECA (Plan, Establish, Collect, and Analyse) process has its’ base in ISO 14598-1:1999 but is developed by the Software Engineering Institute (SEI) and National Research Council Canada (NRC). PECA is flexible and can help organizations in their evaluations, it can also be tailored to fit special type of
needs. The process is as the names implies based on four sub-process and can be used as a guideline when defining and carrying out the evaluation.

3.3.2.1 Plan the evaluation
An evaluation always has a goal and that is to choose the most suitable system for the organization’s needs. To be able to achieve that one must plan the evaluation to be able to answer questions like:

- What is expected with the evaluation?
- How will the evaluators measure success?
- What constraints must the evaluators adhere to?
- How should the resources be spent?

In other words the evaluators must have a clear picture of the scope, goal and delimitations for the evaluation in order to carry through an accurate evaluation with verifiable results. For example the evaluation can have two different approaches “First fit” vs. “Best fit” (“Satisficing” and “Maximizing” can be used in other terminologies). First fit denotes that the first system or product that fulfils the needs is to be chosen and the evaluation can be seen as completed. Best fit however, implies that the group of systems or products that might match the needs will be evaluated first and the decisions will be based on the outcome. As this example illustrates one should always consider the available resources before planning the evaluation. (Comella-Dorda, 2004)

3.3.2.2 Establish the criteria
The defined criteria are the ones that the specific system must fulfil, and must therefore be specified in an appropriate way. One way of defining
criteria is to transform the specified requirements from the RE processes in a suitable manner so that it is possible to measure the evaluated systems. One should however decide which requirements should be transformed into criteria, some requirements specifications can be several documents and some can be just a few pages, it is important to have the time budget in mind. The criteria should also be prioritized to effectively perform the evaluation and enhance the most important ones, but if the requirements are already prioritized then there is no need to reprioritize the criteria, one can just use the same prioritization. If a prioritization is needed one can just use one of the methods described in section 3.2.2.6. (Comella-Dorda, 2004)

However it is often difficult to transform requirements into criteria, requirements are often written at an abstract level to be flexible for multiple solutions while the criteria is supposed to be on a more measurable level. The transformation to criteria should therefore be stated in terms of capabilities. E.g. Requirement “The system shall be easy to use” is too abstract to be measured as a criterion and could therefore be transformed to “The learning curve for the system shall not exceed five days”. (Comella-Dorda, 2004)

3.3.2.3 Collect the data

Collecting accurate data is crucial to obtain a good evaluation of a system. Data collection should not just be accurate it must also be simple and repeatable in order to measure and analyse it. To gather such data one can use different techniques, few of them are slightly described below: (Comella-Dorda, 2004)

- Literature reviews: The technique is as the name implies based of reviewing relevant documents and specifications for the specific product. Though have in mind that the materials from the vendors often enhance the well working parts while the parts that are not so smooth are often omitted.
- Hands-on techniques: Used often to verify claims, determine interactions with other components, determine performance etc. One type of a hands-on technique is use of a prototype, where the major features of the system are evaluated.
- Benchmarking: Is another technique but should be used carefully since most of them are provided by the vendors themselves and are often unreliable.

3.3.2.4 Analyse the data

The collected data is often raw and needs to be consolidated to be able to analyse it. Consolidation of data is as said often used when one has collected raw data that can be directly analysed. Transformation from raw data to information can be performed with different techniques, e.g. All-to-dollars and the more common Weighted aggregation technique. Raw data
has however not been used in this project so there will not be any further discussion on this topic. (Comella-Dorda, 2004)

The collected data still needs to be analysed to ensure adequacy and accuracy and there are few techniques that can be used for this type of analysis.

**Sensitivity Analysis:** This technique is based on analysing how sensitive the collected data are to small deviations. One can then determine the impact of these changes on the overall fitness. This technique can be used if one believes that the collected data are not completely accurate. (Comella-Dorda, 2004)

**Gap Analysis:** This technique is often used to determine the differences between the as-is processes (end-user processes) and the investigated/evaluated processes for the to-be process with the new solution. Usually a Gap analysis starts with a “Yes/No” matrix, i.e. a matrix that provides a broad understanding in if the solutions fulfil the defined criteria, and then continuous in finding and defining the gaps. The Gap result of the analysis is presented in matrix with the different solutions on top row and the criteria downwards. This matrix provides an overview of the different solutions and gives a deeper understanding in how well they fulfil the defined criteria. (Comella-Dorda, 2004)

![Gap Information](image)

Table 5, Illustration of GAP analysis (Comella-Dorda, 2004)

### 3.4 Prototype testing

Prototype tests are a validation method of requirements as described in section 3.2.4.3. A prototype facilitates the understanding of how a function or system will be working in its environment. It makes it also easier to find the things that could be improved. One should though not forget that a prototype is just for demonstration purpose. Prototypes doesn’t present the finished product, a prototype is an initial version of the product, i.e. used to present ideas to the users. (Gottesdiener, 2005) (Alexander, 2002) (Kotonya, 1998)
4 Development of Requirements

The elicited, analysed, specified, validated and managed requirements in this project have been defined with help of different techniques and methods described, cf. section 3.2. This section will describe how the requirements have been developed through these processes.

4.1 Requirements Elicitation

The elicitation phase has been carried out mainly through interviews with stakeholders and studying documents, observations have only been used when investigating the control system physically. The objective with the document study have been to give the requirements engineer basic and background knowledge of how the control system is built and works, but also knowledge of how the business processes looks like. The document study had to be performed before the interviews with the stakeholders to be able to develop relevant questions to the interviewees. The interviews have been the main source when eliciting requirements and have been performed iteratively to cover the gaps in the findings. Following stakeholders have been interviewed to elicit requirements and get an insight in how the control system is built and works as a complement for the document study: Mr. Lars, Mr. Daniel, Mr. Janne and Mr. Alfred. Mr. Stig and Mr. Hans have been interviewed to get a better insight in the components of the control system, GWS respectively OWS/SER. Mr. Erik have been one of the supervisors for the Thesis but resigned, he has been involved in elicitation part and planning part of this project.

Table 6 presents a profiling of the interviewed stakeholders and

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Roles</th>
<th>Responsibilities</th>
<th>Interests</th>
<th>Success Criteria</th>
<th>Concerns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lars</td>
<td>Customer support project manager</td>
<td>Customer support</td>
<td>Streamline customer support</td>
<td>Applicable on all customer sites, Remote connection, Automatic retrieval of log files, Problem/ alarm report</td>
<td>Problem with software update due to laws</td>
</tr>
<tr>
<td></td>
<td>Subject matter expert</td>
<td>Create, modify, plan and close trip cases</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Contact person for customers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daniel</td>
<td>Coordinator</td>
<td>CCB member, Dispatching work to technicians</td>
<td>Streamline interactions and work between business units and customer</td>
<td>Applicable on control system, Version 1, Remote connection</td>
<td>Remote connection through mobile broadband that has high availability all over the world</td>
</tr>
<tr>
<td><strong>Erik</strong></td>
<td><strong>Controller</strong></td>
<td>Controller for delivered control systems</td>
<td>Implementing a product for remote diagnostics of control systems</td>
<td>Remote support of control system File transfer to and from control system Problem/ alarm report</td>
<td>Existing remote connection technology isn’t up to date</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>------------------------------------------</td>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>Janne</strong></td>
<td><strong>Developer</strong></td>
<td>Development of databases for the control system</td>
<td>Remote SW &amp; HW upgrade of control system components</td>
<td>Component/ device version management Remote upgrade of components</td>
<td>Adhering to existing control system protocols</td>
</tr>
<tr>
<td><strong>Alfred</strong></td>
<td><strong>Customer support project manager</strong> <strong>Subject matter expert</strong> <strong>Contact person for customers</strong></td>
<td>Customer support Create, modify, plan and close trip cases</td>
<td>Streamline customer support</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Stig</strong></td>
<td><strong>Subject matter expert (GWS)</strong></td>
<td>GWS specialist in SCADA and control systems</td>
<td>Establish a reliable remote connection</td>
<td>Applicable on all customer sites Remote connection Automatic retrieval of log files Problem/ alarm report</td>
<td></td>
</tr>
<tr>
<td><strong>Hans</strong></td>
<td><strong>Subject matter expert (OWS/SER)</strong></td>
<td>OWS/SER specialist</td>
<td>Remote service of control system</td>
<td>Reliable remote connection</td>
<td></td>
</tr>
</tbody>
</table>

| 4.2 Requirements Analysis |

The analysis of the requirements has been performed with Process Maps, Context diagrams and Prioritization of requirements. The main business processes for the customer support includes Trip management and Upgrading hardware and software. There is also another process on-going parallel to the two processes viz. Change Control Board process. The business processes can be described with different models and the modelling language used for this is Business Process Modelling Notation (BPMN). This modelling language provides a notation that is readily understandable by all business users and it results in Business Process Diagrams (BPD). The BPDs for the different business process can be found in the appendix chapter.
• **Trip management:** A problem or fault that occurs on a customer site is called a *trip*. The trip handling today works in that way that the customer contacts the customer support for help. Customer support needs log files from customer site in order to help and solve the trip, and retrieving the right log files is the main problem. The log files are located on different computers in the control system and contain information on every occurred event. Usually the customer needs to extract these log files by guidance from customer support, and it occurs often for situations where the customer doesn’t have the knowledge to be able to extract these files. This type of a function with direct communication link with customer sites will prevent these types of problem and will reduce the time spent on retrieving log files. Customer support will be able to manually or automatically retrieve log files when a trip occurs and will be prepared when the help is needed.

• **Upgrade due to manufacturing fault:** One type of an upgrade is because of detected manufacturing faults or software faults. The situation here is that e.g. a series of flash disks, needs to be replaced due to manufacturing fault. In these types of situations usually the replacement is performed by a service engineer from a local office or in some cases a service engineer from the headquarters in Sweden. This function will provide the ability to assist the service engineer on site from office in Sweden instead of having the e.g. technical specialist also on site, which will results in higher expenses. But also in situation where the customer itself have the ability to perform the replacement and only needs assistance for software upgrade on the component.

• **Change Control Board:** This process is parallel to these two processes and the *Change Control Board*’s tasks are to follow and keep track of these trips, changes and faults. The process starts by reviewing incoming cases, the cases and their descriptions are located in the Case Database, followed by handling and coordinating them, cf. Appendix C for involved roles in the process. All these trips, changes and faults that are affecting projects are compiled in a matrix, for example:

<table>
<thead>
<tr>
<th></th>
<th>Fault A</th>
<th>Fault B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Project B</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Project C</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Table 7, Example of a project matrix*

With this matrix the CCB can get a better overview of the trips and their effects on different sites/projects and can determine upcoming actions. For example if a base solution needs to be changed, the CCB takes decisions and follows the process for making the changes so that new
projects are updated. For old projects preventive activities will be performed.

A context diagram illustrates what inputs the function needs to provide outputs and how external entities interact with the function in its environment. The core thing a context diagram illustrates is how these external entities provide and receive information to and from the system. The context diagram is usually just one diagram but in this project it has been divided into two diagrams to make the diagrams more clear.

- The first diagram illustrates what types of log files the company requires for trip management and upgrade scenarios and wherefrom these log files can be retrieved. The log file collection process will be performed by the users at the company, who will be able to connect to site and transfer the needed log files from their corresponding places to the used computer at the company. The customer will not need to assist the service engineer for file collection tasks.

- The other diagram is a simple illustration of how the different component’s software of customer’s control system can be upgraded. By establishing a connection to customer’s site one will be able to perform software upgrades with this function. The software upgrades will consist of transferring new files to their corresponding places in the MACH2 control system and running the upgrade script. The new files will be sent out from the company and a local service engineer or someone from the headquarters will perform the file transfer tasks. The context diagrams can be found in the appendix chapter.

Prioritization of requirements is a very important process and by prioritizing requirements one is able to get an overview of important functionalities that the function needs to have. If the evaluation of the function doesn’t comply with the requirements then the function is not the right one and cannot support the business in its processes described above. There are several techniques to use for prioritization and the one chosen for this specification is a graded letter scale from A to D, where A is the most critical requirement. (Wieger, 2003 pp. 247-258)

The description for each number is described below:

A. Implies the most important requirement, highest ranked
B. High priority, it’s not the system’s most core functionalities but still needed to support the processes
C. Medium priority, not so important but it would still ease off the work in the business processes
D. Low priority, this requirement implies useful functionalities but not necessary for the business process

The following group members performed the prioritization of the requirements in a focused meeting in this project: Mr. Lars, Mr. Daniel, Mr. Alfred and Mr. Sel.
4.3 Requirements Specification

Requirements are often divided into three types of requirements:

- **Functional requirements**: are requirements that define what the function is supposed to accomplish. The functional requirements describe the captured use cases in the analysis phase, e.g. in expressed form of “system shall do …”. Functional requirements define the application architecture. (Gottesdiener, 2005) (Kotonya, 1998)

- **Non-functional requirements** (so called Quality attributes): define the technical architecture, i.e. quality attributes such as design, performance, reliability, security, safety etc. These requirements support the functional requirements and they place restrictions on the development and specify external constraints on the implementation. (Gottesdiener, 2005) (Kotonya, 1998)

- **External interface requirements**: describes the requirements coming from external interfaces. Example of interfaces that may exist is listed below:
  - Software interfaces: if the system will have to interface some other software, such as other IT-systems, this will be described here
  - Hardware interfaces: hardware such as computers, network cables, routers, etc. may interface the new function

The developed requirements in this project are presented in sections below and is sorted according to corresponding processes for the function (communication, information, maintenance work) and not divided into functional, non-functional and external interface requirements. If one recalls the scope of the project, which was to define requirements and get better insight in what kind of a solution one should implement, and not a procurement project where every detail must be defined, the approach must be on a high level. This yields a broad perspective of the desired and needed product that has come to hand in the evaluation. If one should have defined requirements on a more detailed level the requirements specification and the evaluation of the products would have taken much more time and effort than this projects had at its disposal.

### 4.3.1 Communication

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirements</th>
<th>Comment</th>
<th>Dependencies</th>
<th>Related</th>
<th>Priority</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The function shall be applicable on all type of MACH2 control systems, all around the world</td>
<td>Microsoft windows environment for MACH2 control system. Some sites doesn’t even have internet connection outwards.</td>
<td></td>
<td></td>
<td>A</td>
<td>Mr. Lars</td>
</tr>
<tr>
<td>1.2</td>
<td>The function itself must be an external unit</td>
<td>1.1</td>
<td>A</td>
<td>Mr. Lars</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>The function shall support Version 1 of the control system</td>
<td>As from 2006/2007</td>
<td>1.1</td>
<td>A</td>
<td>Mr. Lars, Mr. Daniel</td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>The function shall be able to support Version 2 of the control system</td>
<td>Version 2 is the upcoming control system and is under development</td>
<td>1.1</td>
<td>C</td>
<td>Mr. Lars, Mr. Daniel, Mr. Erik</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>The function shall support Version 0 of the control system</td>
<td>2006 and before (approximately 10 years)</td>
<td>1.1</td>
<td>C</td>
<td>Mr. Lars</td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>The function shall be prepared to have different user levels for access control</td>
<td>E.g. admin and user level</td>
<td></td>
<td></td>
<td>Mr. Daniel</td>
<td></td>
</tr>
<tr>
<td>1.7</td>
<td>User shall be able to connect remotely to a site that has this function</td>
<td>Each site has a unique ID</td>
<td>1.1, 1.2</td>
<td>A</td>
<td>Mr. Lars</td>
<td></td>
</tr>
<tr>
<td>1.8</td>
<td>The function shall support communication through…</td>
<td>Communication possibilities to connect to customer site remotely from any company office</td>
<td>1.1, 1.3</td>
<td></td>
<td>Mr. Lars, Mr. Daniel</td>
<td></td>
</tr>
<tr>
<td>1.8.1</td>
<td>telephone line</td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.8.2</td>
<td>ethernet</td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.8.3</td>
<td>mobile communication</td>
<td>The mobile communication link must have high availability regardless the geographical location and use common reliable technology like GSM, WCDMA or LTE</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.8.4</td>
<td>satellite</td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9</td>
<td>The function shall support protocol…</td>
<td>1.3, 1.4, 1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9.1</td>
<td>IEC-61850</td>
<td></td>
<td>D</td>
<td>Mr. Janne</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9.2</td>
<td>OPC</td>
<td></td>
<td>D</td>
<td>Mr. Erik</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9.3</td>
<td>SNMP</td>
<td></td>
<td>D</td>
<td>Mr. Erik</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.9.4</td>
<td>DNP</td>
<td>GWS and protection devices</td>
<td>D</td>
<td>Mr. Janne</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 4.3.2 Information

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirements</th>
<th>Comment</th>
<th>Dependencies</th>
<th>Related</th>
<th>Priority</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>User shall be able to retrieve log files from the control system manually…</td>
<td>Each component in the control system has a version</td>
<td>1.7, 1.8</td>
<td>B</td>
<td></td>
<td>Mr. Lars, Mr. Daniel</td>
</tr>
<tr>
<td>2.0.1</td>
<td>User shall be able to retrieve software and hardware version information on components</td>
<td>Statistical purpose, but what kind of information is yet to be decided</td>
<td>2.0</td>
<td>C</td>
<td></td>
<td>Mr. Erik</td>
</tr>
<tr>
<td>2.1</td>
<td>The function shall be able to cyclically retrieve log files from control system</td>
<td></td>
<td>1.3, 1.8</td>
<td>2.0</td>
<td></td>
<td>Mr. Lars</td>
</tr>
<tr>
<td>2.2</td>
<td>User shall be able to select which log files to cyclically retrieve</td>
<td></td>
<td>1.3, 1.7, 1.8</td>
<td>C</td>
<td></td>
<td>Mr. Lars</td>
</tr>
<tr>
<td>2.3</td>
<td>The function shall be able to retrieve log files from the control system's components…</td>
<td></td>
<td>1.3, 1.7, 1.8, 2.0</td>
<td>2.0</td>
<td></td>
<td>Mr. Lars, Mr. Daniel</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Main computers (MACH2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>2.3.2</td>
<td>OWS/SER computer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>2.3.3</td>
<td>GWS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>2.3.4</td>
<td>SMC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>2.4</td>
<td>The function shall be able to retrieve log files of unlimited size</td>
<td>Email and sms, but the criteria for when a notification should be sent is yet to be decided</td>
<td>1.3, 1.7, 1.8, 2.0, 2.3</td>
<td>A</td>
<td></td>
<td>Mr. Lars, Mr. Daniel</td>
</tr>
<tr>
<td>2.5</td>
<td>The function shall be able to support email/sms notification when a trip occurs</td>
<td></td>
<td>1.3, 1.7, 1.8, 2.0, 2.3</td>
<td>C</td>
<td></td>
<td>Mr. Lars</td>
</tr>
<tr>
<td>2.6</td>
<td>User shall be able to change destination for email/sms notification</td>
<td></td>
<td>1.6</td>
<td>2.5</td>
<td></td>
<td>Mr. Lars</td>
</tr>
</tbody>
</table>

### 4.3.3 Maintenance work

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirements</th>
<th>Comment</th>
<th>Dependencies</th>
<th>Related</th>
<th>Priority</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>The function shall support remote file transfer to control system components for version…</td>
<td></td>
<td>1.3, 1.7, 1.8</td>
<td></td>
<td></td>
<td>Mr. Lars, Mr. Daniel</td>
</tr>
<tr>
<td>3.0.1</td>
<td>Version 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>3.0.2</td>
<td>Version 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>
4.4 Requirements Validation

Validation of requirements was performed through peer reviews. They are often performed by a small group of stakeholders, but could for some cases be performed together with interviewed interviewees, if nothing else are decided. For this project it was often a group of 2-4 people and subsets of the requirements was validated first in order to review everything together. The validation of the developed maps and diagrams for the analysis phase was also performed by peer reviews.

Operational prototype is another technique for validating requirements but could not be performed in this project, please see section 6.3.

4.5 Requirements Management

Management of requirements is an on-going phase performed parallel to the other RE processes and includes monitoring and controlling the requirements for statuses and changes. The management handling was mainly performed by the requirements engineer but included also the supervisors for the project. The requirements engineer together with the two supervisors acted as a CCB. All changed requirements and their statuses should have been documented in a Requirements Trace Matrix (RTM). RTM was another technique that was intended to be used but didn’t come in handy since the prototype testing wasn’t implemented. Usually the requirements are changed a little bit when prototype testing begins, this is often caused when realising incorrect specification of the requirements.
5 Evaluation and results

The PECA method has been the base for the evaluation of the two solutions, i.e. plan the evaluation, establish the criteria, collect the data and analyse the data. The evaluation has been planned early in this project and before the evaluation was started following questions needed to be were answered:

- What is expected with the evaluation?
  The outcome of the evaluation is to find a suitable solution that matches the specified requirements in order to begin testing an operational prototype.
- How will the evaluators measure success?
  The success will be measured by investigating how well the solutions comply with the specified criteria and requirements using evaluation techniques like Gap analysis.
- What constraints must the evaluators adhere to?
  The time budget for the evaluation phase is a very important constraint the evaluator must adhere to. The determined time plan must be fulfilled in order to complete the set goals for the project.
- How should the resources be spent?
  The resources should be spent as planned in the beginning for the project. The evaluator should prioritize a thorough evaluation of the two solutions before carrying on to the prototype or market research phase.

Having these questions answered gives one the scope, goal and delimitations for the evaluation, but the answers for the last two questions are a bit contradicting. The meaning with the answers is that the evaluation must be performed thoroughly in order to continue to the next phase in the project. If not, then it will raise problems when evaluating and testing the prototype, because one will probably notice deficiencies in the products. Clearing this gives us the scope for the evaluation:

Match the solutions to the specified requirements within given time frame but make the evaluation thorough.

Establishing criteria for the evaluation have not been performed exactly as the PECA process but the process has been the base. The difference is that the requirements have directly been used as the criteria in the evaluation instead of rephrasing the requirements into criteria, cf. section 3.3.2.2. It was though planned in the beginning that the requirements that wasn’t going to be matched to the solutions, should instead been notified and evaluated in the prototype phase.

Like other evaluations the intention with this evaluation was not to perform measurements in order to collect and analyse data. The evaluation was instead based on finding matches and gaps between the specified requirements and the solutions. The purpose of the evaluation was to find a suitable solution to be able to test the different work tasks on a prototype.

The evaluation started by studying documents for the two solutions. The purpose of the documents were to get an insight in how the solutions work, how they are built, what
requirements they have to function and what features they have. The documents consisted mainly of:

- presentation material
- product overview, and
- product specifications

The “Best fit” (“Maximizing”) approach was used in the evaluation, this approach implies that all the solutions will be evaluated and then analysed before taking any decisions. The depth of the evaluation was decided to be volatile, cf. section 2.2, i.e. since the evaluation started by evaluating the A-graded requirements first one could not now beforehand if one should continue the evaluation with the next level (B-graded requirements) or to not. The two solutions were evaluated for the A-graded requirements first, “Best-fit” approach, and then reviewed to determine if one solution weight over the other, in order to continue with the evaluation or not.

As described above there was no collection of data, finding matches together with Gap analysis was performed instead. The findings were first notified by Yes/No values, “Yes” if the solution matches the requirement and “No” if not. These values were then studied in form of a Gap analysis to be able to analyse the findings and find the gaps/deficiencies.

Table 8 presents the complete table of the Yes/No matrix with the Gap analysis and will be discussed in the next section.

<table>
<thead>
<tr>
<th>Priority</th>
<th>ID</th>
<th>Requirements</th>
<th>Yes/No matrix with Gap analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.1</td>
<td>The function shall be applicable on all type of MACH2 control systems, all around the world</td>
<td>No, existing internet connection is required</td>
</tr>
<tr>
<td>A</td>
<td>1.2</td>
<td>The function itself must be an external unit</td>
<td>No, just a software application (Java)</td>
</tr>
<tr>
<td>A</td>
<td>1.3</td>
<td>The function shall support Version 1 of the control system</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>A</td>
<td>1.7</td>
<td>User shall be able to connect remotely to a site that has this function</td>
<td>Yes, but requires existing connection to site</td>
</tr>
<tr>
<td>A</td>
<td>1.8</td>
<td>The function shall support communication through</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>1.8.2</td>
<td>Ethernet</td>
<td>Yes, existing connection is required</td>
</tr>
<tr>
<td>A</td>
<td>1.8.3</td>
<td>Mobile communication</td>
<td>No, needs to be complemented, cf. section 6.2.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>A</td>
<td>2.4</td>
<td>The function shall be able to retrieve log files of unlimited size</td>
<td>Yes</td>
</tr>
<tr>
<td>B</td>
<td>2.0</td>
<td>User shall be able to retrieve log files from the control system manually</td>
<td>Yes, both manual and automatic possibility</td>
</tr>
<tr>
<td>B</td>
<td>2.3</td>
<td>The function shall be able to retrieve log files from the control system's components</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>2.3.1</td>
<td>Main computers (MACH2)</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>2.3.2</td>
<td>OWS/SER computer</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>2.3.3</td>
<td>GWS</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>2.3.4</td>
<td>SMC</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>3.0</td>
<td>The function shall support remote file transfer to control system components for version</td>
<td>Yes, file transfer is supported but should be verified</td>
</tr>
<tr>
<td>B</td>
<td>3.0.1</td>
<td>Version 1</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>3.1</td>
<td>User shall be able to transfer files to control system's components</td>
<td>Yes, file transfer is supported but should be verified</td>
</tr>
<tr>
<td>B</td>
<td>3.1.1</td>
<td>Main computers (MACH2)</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>3.1.2</td>
<td>OWS/SER computer</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>3.1.3</td>
<td>GWS</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>3.1.4</td>
<td>SMC</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>3.1.5</td>
<td>Firewall</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>3.1.6</td>
<td>Switch</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>B</td>
<td>3.2</td>
<td>The function shall be able to transfer log files of unlimited size</td>
<td>Yes</td>
</tr>
<tr>
<td>C</td>
<td>1.4</td>
<td>The function shall be able to support Version 2 of the control system</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>C</td>
<td>1.5</td>
<td>The function shall support Version 0 of the control system</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>C</td>
<td>1.6</td>
<td>The function shall be prepared to have different user levels for access control</td>
<td>Yes</td>
</tr>
<tr>
<td>C</td>
<td>1.8.1</td>
<td>Telephone line</td>
<td>Yes</td>
</tr>
<tr>
<td>C</td>
<td>1.8.4</td>
<td>Satellite</td>
<td>No information</td>
</tr>
<tr>
<td></td>
<td>2.0.1</td>
<td>User shall be able to retrieve software and hardware version information on components</td>
<td>Prototype testing needed</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>C</td>
<td>2.1</td>
<td>The function shall be able to cyclically retrieve log files from control system</td>
<td>Yes, but criteria needs to be decided, cf. section 6.1</td>
</tr>
<tr>
<td>C</td>
<td>2.2</td>
<td>User shall be able to select which log files to cyclically retrieve</td>
<td>Yes, but criteria needs to be decided, cf. section 6.1</td>
</tr>
<tr>
<td>C</td>
<td>2.5</td>
<td>The function shall be able to support email/sms notification when a trip occurs</td>
<td>Needs to be verified</td>
</tr>
<tr>
<td>C</td>
<td>2.6</td>
<td>User shall be able to change destination for email/sms notification</td>
<td>Needs to be verified</td>
</tr>
<tr>
<td>C</td>
<td>3.0.2</td>
<td>Version 0</td>
<td>Yes, file transfer is supported but should be verified</td>
</tr>
<tr>
<td>C</td>
<td>3.0.3</td>
<td>Version 2</td>
<td>Yes, file transfer is supported but should be verified</td>
</tr>
<tr>
<td>D</td>
<td>1.9</td>
<td>The function shall support protocol</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1.9.1</td>
<td>IEC-61850</td>
<td>Needs to be verified</td>
</tr>
<tr>
<td>D</td>
<td>1.9.2</td>
<td>OPC</td>
<td>Needs to be verified</td>
</tr>
<tr>
<td>D</td>
<td>1.9.3</td>
<td>SNMP</td>
<td>Needs to be verified</td>
</tr>
<tr>
<td>D</td>
<td>1.9.4</td>
<td>DNP</td>
<td>Needs to be verified</td>
</tr>
</tbody>
</table>

Table 8, Evaluation matrix: Yes/No together with Gap analysis
6 Analysis
In this section the requirements engineering, evaluation and prototype is analysed.

6.1 Requirements Engineering

Requirements define a function of a system or its components. Product functions can be described as inputs and outputs, and in some cases as behaviours. Requirements define what a function, application or system is supposed to accomplish. The specified requirements will be the base for the system design and has been structured and written in a projection point of view. This view implies that each requirement has been written from a functionality or user perspective. The requirements have also been checked with regard to the criteria below of how one should specify requirements. (Alexander, 2002 pp. 96-107) (Davis, 1993 pp. 15-31):

- Correctness
- Unambiguous
- Completeness
- Consistent
- Prioritized
- Verifiability
- Modifiability
- Traceability

The developed framework for the RE processes have been carried out with help of chosen techniques and methods, cf. section 2.2 and 4. The chosen methods turned out to be of great use in order to perform the first part of this project, i.e. a Requirements Specification. As mentioned the requirements engineer did not have any knowledge of what kind of a product the company requested for their problem or how the control system, where the product would be placed, operates.

The outcome of the RE was that the intended product should fulfil the specified requirements. Following is a summary of the requirements that can be seen as main criteria, and the products must fulfil these in order to be accepted as a possible solution:

- The product must be compatible with the existing control system (Version 1), i.e. compatible with Microsoft windows environment
- It must be an external unit, i.e. no program is allowed to run on the existing components in the control system
- Support remote access, i.e. the product must have an embedded wireless connection that can be accessible
- Support file transfer to and from the control system’s different components
- Monitoring of the control system, i.e. embedded feature that can monitor the different components in the control system

As the problem is to set up a communication link towards customer’s site in order to monitor and support the control system the solution must become a part of the control system and interact with it. The existing control system, Version 1, has several components that has Microsoft Windows environment.
The control system has also a previous Version 0, and an upcoming Version 2. Considering this the intended solution should primarily be compatible with existing system, but also be open for modifications to be able to support the previous and upcoming versions. Finding these possibilities in a solution could be difficult to find and is more likely to be negotiated in terms with the vendors rather than stating them as requirements.

Possible solutions must however be an external unit, i.e. the program must run on a separate device that will become a part of the control system. The solutions must have an embedded remote access platform that supports wireless broadband and be compatible with the most common telecommunication frequencies. Since the control systems are delivered all around the world, where some don’t even have a communication link outwards, it requires a wireless solution that covers every existing control system that are in range of coverage.

The company needs several files from a control system in order to support when a problem occurs. These files are located in different parts of the control system and needs to be transferred. The solution should therefore not only have remote access, it should also be able to locate and transfer these files automatically or manually by the user. This type of requirements however raises difficulties in the requirements specification, e.g. if a solution should have an automatically feature for which criteria a message should be sent out for a service engineer in monitoring and support purpose? Defining these criteria for this type of a requirement is very difficult and requires more effort and involvement from every involved party, i.e. people from different business units and the customers.

### 6.2 Evaluation

The analysis of the evaluation has been derived from the PECA process but has been adapted to fit this projects time budget and resources. PECA recommends two techniques for analysis of data, Sensitivity and Gap analysis, see section 3.3.2.4. Sensitivity analysis is supposed to be used if one has collected raw data that could lead to uncertainty. The uncertainty often arises if one have measured values or e.g. made assumptions concerning weights in the weight aggregation. Therefore considering the project scope and the extent Gap analysis has been the main technique in this project. Below is a summary of the Gap analysis for the main criteria the solutions needed to fulfil, cf. section 4.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C1: The product must be compatible with the existing control system (Version 1), i.e. compatible with Microsoft windows environment</td>
<td>Microsoft windows based software</td>
<td>Proprietary software/Operating system</td>
<td></td>
</tr>
</tbody>
</table>
C2: It must be an external unit, i.e. no program is allowed to run on the existing components in the control system. Only a software

C3: Support remote access, i.e. the product must have an embedded wireless connection that can be accessible. Requires an existing connection

C4: Support file transfer to and from the control system’s different components. Manual transfer but has possibility for automatic transfer

C5: Monitoring of the control system, i.e. embedded feature that can monitor the different components in the control system. Monitoring and alarm possibilities

<table>
<thead>
<tr>
<th>Feature</th>
<th>Requirement</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>It must be an external unit, i.e. no program is allowed to run on the existing components in the control system</td>
<td>Only a software</td>
</tr>
<tr>
<td>C3</td>
<td>Support remote access, i.e. the product must have an embedded wireless connection that can be accessible</td>
<td>Requires an existing connection</td>
</tr>
<tr>
<td>C4</td>
<td>Support file transfer to and from the control system’s different components</td>
<td>Manual transfer but has possibility for automatic transfer</td>
</tr>
<tr>
<td>C5</td>
<td>Monitoring of the control system, i.e. embedded feature that can monitor the different components in the control system</td>
<td>Monitoring and alarm possibilities</td>
</tr>
</tbody>
</table>

Table 9, Summary of Gap analysis for the main features

6.2.1 Evaluation of P1, Remote Access Platform

If we summarize Table 9 for P1 we can see that P1 fulfils three of the main required features, P1 however doesn’t fulfil criteria C2 and C3. P1 is a program that’s supposed to be installed on a computer in order to setup the different devices one wants to monitor in the control system. Since P1 is a program that requires an already existing Internet connection, to function in the control system, it lacks a very essential feature for the company’s purpose.

P1 can however be complemented so C2 and C3 are fulfilled, i.e. to fit the required features for a solution. For that P1 needs to be complemented with:

- an external computer,
- a computer that has an embedded wireless modem or at least supports it (if the computer doesn’t have wireless modem embedded it can always be complemented from a third party supplier), and
- arrange with a network operator for a worldwide wireless connection that covers almost every existing control system

6.2.2 Evaluation of P2, Remote Service

P2 however fulfils the main features as a solution except C1. P2 is developed by a business unit within the company for their specific needs, and is a proprietary software/operating system. P2 does overall fit the specified requirements but the proprietary software/operating system requires a deeper investigation in how difficult it could be to adapt it for this kind of a control system.
To be able to choose P2 as a solution one must first investigate:

- How difficult it is to adapt the software/operating system?
- How much time could it take?
- How much could it cost?
- Who is supposed to develop the new software?
- Who will support it once it is implemented?

### 6.3 Prototype

The plan and a sub-goal for this project was if one of the existing solutions, after evaluation of them, should match the specified requirements a prototype for that solution should be implemented. The prototype should be used as a hands-on technique, cf. section 3.3.2.3, to test the product in its working environment. These tests could include evaluations like: which outputs are generated for a specific input, how the monitoring of control system components does work, and how does the file transfer work etc. The prototype phase could however not be performed due to problems with establishing contact and support for the testing and implementation with the vendor that ran over time.
7 Discussion

Hevner et al. has derived seven guidelines from fundamental principle in design-science. These guidelines describe how one should/could perform the scientific research in order to keep the scientific quality high. If one could fulfil these guidelines (can also be seen as criteria), one could then be assured that the developed framework and written report for this project keeps a good quality for design science. These seven guidelines can be found in Table 10, and this section of the report will discuss how this project follows these guidelines/or fulfils these criteria. (Hevner, et al., 2011)

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Guideline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Design as an Artefact</td>
<td>The research must produce a viable artefact in the form of a construct, a model, a method, or an instantiation.</td>
</tr>
<tr>
<td>2.</td>
<td>Problem Relevance</td>
<td>The objective of the research is to develop technology-based solutions to important and relevant business problems.</td>
</tr>
<tr>
<td>3.</td>
<td>Design Evaluation</td>
<td>The utility, quality, and efficacy of a artefact must be rigorously demonstrated via well-executed evaluation methods.</td>
</tr>
<tr>
<td>4.</td>
<td>Research Contributions</td>
<td>Effective search must provide clear and verifiable contributions in the areas of the artefact, foundations, and/or methodologies.</td>
</tr>
<tr>
<td>5.</td>
<td>Research Rigor</td>
<td>Research relies upon application of rigorous methods in both the construction and evaluation of the artefact.</td>
</tr>
<tr>
<td>6.</td>
<td>Design as a Search Process</td>
<td>The search for an effective artefact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.</td>
</tr>
<tr>
<td>7.</td>
<td>Communication of Research</td>
<td>Research must be presented effectively both to technology-oriented as well as management-oriented audiences.</td>
</tr>
</tbody>
</table>

Table 10, Design Science Research guidelines (Hevner, et al., 2011)

Guideline 1

This guideline describes that the research must produce a viable artefact and for this project this will be the developed framework. The developed framework, including the RE and the evaluation phase has been derived from several scientific references. The framework can be seen a set of methods that have been used to perform the different tasks, e.g. interviews, document study, focused meetings etc. These methods have in their turn lead to several maps and models, i.e. process maps, context diagrams etc., that have been used to present the findings. The result from the first part in the developed framework is that a Requirements Specification has emerged. This Requirements Specification has been the base for the evaluation of the two possible solutions that the company had in their mind.

The second part of the framework is the evaluation. In order to develop the PECA process, the SEI has used scientific references and studies. The defined process is supposed to be tailored to fit the evaluation problem in order to support the organization in making carefully and reasoned decisions. The PECA process has therefore been adapted to fit this organizations evaluation.

Thirdly there is a concrete document, viz. the Requirements Specification.

Guideline 2

This project does not include developing of a technology-based solutions, this project’s objective has instead been to develop a framework that will emphasize the business’s
needs and demands, finds a solution that fits these and solves the identified and important problem.

If one now recalls the findings in the analysis of the evaluation, cf. section 6.2:
- P1 must to some extent be complemented and
- P2 must be adapted

in order to fit the desired solution. Solving these problems with the solutions can be seen as developing a technology-based solution.

The prototype implementation in this project should have been where one would have solved the related problems to the solutions.

**Guideline 3**

Guideline 3 describes that the utility, quality and the efficacy must be rigorously demonstrated via well-executed evaluation methods (Hevner, et al., 2011). This means that the intended product must be evaluated not only beforehand, as in the evaluation phase, but also in its environment, i.e. integration and implementation of the solution in its technical infrastructure and business environment. Hevner et al. has listed few evaluation methods for this, three of these methods could and have been applied for this project:

- **Analytical**
  - *Static Analysis*: Examines the structure of the solution for static qualities, e.g. complexity.
  - *Architecture Analysis*: Studies how the product fits into the technical architecture.

- **Experimental**
  - *Controlled Experiment*: Studies the product in a controlled environment for qualities, e.g. usability.

- **Testing**
  - *Functional (Black Box) Testing*: Testing of product to discover failures and identify defects.

If we first look at the analytical evaluation methods, each solutions structure and how the product fits into the control system have been analysed. The static analysis has been performed when determining how well the solutions fit to the specified structural requirements, e.g. the product must be able to fit all the MACH2 control systems (Microsoft windows environment) etc. The architectural analysis has been studied when evaluating the requirements that specify the architecture for the solution, e.g. the product must be an external unit. These types of analysis can be found in Analysis section, cf. section 6.

In the evaluation phase in this project there were some requirements that could not be evaluated, cf. section 5. These requirements as stated needed to be tested and verified in a prototype to be able to determine and decide if the product fulfilled the requirement or not. This prototype phase would have used the experimental and testing evaluation methods, compare with section 3.4 and 6.3.

**Guideline 4**

Every research must provide clear contributions in the areas of the related subjects in order to count as a scientific research (Hevner, et al., 2011). The ultimate assessment is to be able to provide with new and interesting contributions to the science. Hevner et al. discusses and divides the contributions into three categories (Hevner, et al., 2011):
- **The artefact**: If the artefact enables a solution for unsolved problems or can be applied in new and innovative ways on existing knowledge, it can then be seen as contributions to the science.

- **Foundations**: If the knowledge base is extended or improved for appropriately evaluated models, methods etc. it can then be seen as contributions.

- **Methodologies**: If creative development and use of evaluation methods (e.g. analytical, experimental and testing) provides new evaluation metrics, it can also be seen as contributions.

If we look at the artefact as the solution for solving the organizations problem it cannot really be seen as a contribution. There are no facts that the solution for the company’s problem has not been solved in other contexts, one example that can be used as a fact, is that another business unit already use one of the solutions. If the prototype implementation of e.g. P2 can be seen as performed, and we have a finished product that is adapted to the MACH2 control system, we might then see the solution as a contribution because of its uniqueness.

If we see the developed framework for this kind of projects as an artefact, we can then see the developed framework as a contribution. The framework is adapted for this project with the specific chosen techniques and methods in the RE and evaluation phase, but could be generalized to similar projects that include RE for a desired solution/product/system and the evaluation of them with some adaptation to the techniques and methods. One must always review the possibilities for each RE technique and method before deciding which one to carry through.

The outcome of the chosen techniques and methods in the RE and evaluation turned out to be very good. These findings, i.e. use of the techniques and methods, can be more seen as extensions of the knowledge base rather than improving it. For example there is already knowledge that the elicitation technique “interviews with stakeholders” is a good start for elicitation of requirements. This project has also resulted in same conclusion and could therefore be seen as an extension of the knowledge base.

The adapted evaluation method of the PECA process can be seen as a new development of an evaluation method. Therefore it can be seen as an existing method but that has not been evaluated properly.

**Guideline 5**

Guideline 5 describes that the research relies upon the application of rigorous methods in both the construction and the evaluation of the artefact. (Hevner, et al., 2011)

Again, if we see the framework as the artefact, we can see it as a contribution. The construction of the framework could be seen as a new construction tailored with help of different RE techniques and evaluation methods. The RE methods and evaluation methods could be seen as rigorous ones, and they are highly recommended and has also been used rigorously before in projects. The evaluation of them is something that are discussed fluently in this report, e.g. cf. section 2.2, 6 and 7.

If the solution for the problem is the artefact: then it could not really be seen as fulfilled since no prototype has been implemented, i.e. no construction and evaluation have been made on the final product.

**Guideline 6**

Hevner et al. says:
“A problem can be defined as the differences between a goal state and the current state of a system. Problem solving can be defined as a search process using actions to reduce or eliminate the differences.” (Hevner, et al., 2011, p. 85)

The search process to find the desired solutions could in this project be seen as:

- first to reduce the differences between the existing control system and the goal state, with the complemented solution, of the control system, and
- second to eliminate the differences.

The intended prototype implementation in this project was to experiment and test the main functionalities first in order to advance for deeper experimentation and testing of the solution. This project has however so far been able to search and find the desired and effective solution with help of the developed and designed framework. The outcome has so far been as desired of the project, with exception for the prototype phase, so this guideline could also be seen as followed.

**Guideline 7**

Hevner et al. describes that technology-oriented audiences “need sufficient detail to enable the described artefact (solution) to be implemented and used within appropriate organizational context”. The practitioners will try to take advantage of the benefits and the researchers will try to build and extend their knowledge base. The repeatability of the project is one main factor that needs to be taken into consideration when writing the report and this report has however already included this aspect. The management-oriented audiences need however sufficient information to determine if one should commit in purchasing this kind of a solution for their organizational context.

This report will be the presentation of the entire project or Thesis and the author of this report has tried to effectively present each chapter of this report with technology- and management-oriented audiences in mind.
8 Conclusions

One possible problem with the chosen techniques is that there may be important factors that have not been considered. The chosen techniques and methods for the framework could perhaps been developed differently since there is limited time to spend on this project. For example techniques like Facilitated workshops could perhaps have been used, cause they identify conflicting requirements quicker and better. (Gottesdiener, 2005)

Another conclusion is that since there were no knowledge and experience in how the control system operates the outcome for the time plan and execution may have differed a little bit and more time could have been spent on the RE, evaluation and the prototype implementation. This kind of project often requires that the involved persons has background information of e.g. what components the control system is built on and how they interact with each other, to be able to understand where the elicited requirements come from and what kind of a problem there is to solve.

The RE processes has shown that all stakeholders are unified when it comes to what kind of solution they want. However one important problem could be identified during the RE process and that is to determine criteria when someone should be notified of a problem in the control system. Deciding these criteria doesn’t just involve the stakeholders it also involves the executives and must therefore be discussed further in the organisation.

The requirements engineer in this project, Mr. Sel, didn’t have any knowledge at all of the two existing solutions. However, there were few stakeholders that had been investigating the existing solutions on beforehand. This is a common problem where people can be influenced by their own impressions in order to give biased information. This problem have not really affected the outcome of this project, as discussed earlier all the involved stakeholders where unified in what kind of a solution they needed and a neutral party has performed the evaluation.

Regarding the conclusions on what solution the company should choose is to first read sections 6.2.1 and 6.2.2 before taking any decisions. The author’s conclusion is to start with investigating the bullets in section 6.2.2 for P2 first. This conclusion have been derived and been based on the evaluation and the analysis of the two products.
9 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVS</td>
<td>Anti-Virus Staging computer</td>
</tr>
<tr>
<td>BPD</td>
<td>Business Process Diagram</td>
</tr>
<tr>
<td>BPMN</td>
<td>Business Process Modelling Notation</td>
</tr>
<tr>
<td>CCB</td>
<td>Change Control Board</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial-Of-The-Shelf</td>
</tr>
<tr>
<td>GWS</td>
<td>GateWay Station</td>
</tr>
<tr>
<td>HMI</td>
<td>Human Machine Interface</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineering</td>
</tr>
<tr>
<td>NRC</td>
<td>National Research Council Canada</td>
</tr>
<tr>
<td>OWS</td>
<td>Operator WorkStation</td>
</tr>
<tr>
<td>PECA</td>
<td>Plan, Establish, Collect and Analyse</td>
</tr>
<tr>
<td>RE</td>
<td>Requirements Engineering</td>
</tr>
<tr>
<td>RTM</td>
<td>Requirements Trace Matrix</td>
</tr>
<tr>
<td>SC</td>
<td>Series Compensator</td>
</tr>
<tr>
<td>SCADA</td>
<td>Supervisory Control And Data Acquisition</td>
</tr>
<tr>
<td>SEI</td>
<td>Software Engineering Institute</td>
</tr>
<tr>
<td>SER</td>
<td>Sequence of Event Recorder</td>
</tr>
<tr>
<td>SMC</td>
<td>Security Management Console</td>
</tr>
<tr>
<td>SVC</td>
<td>Static VAr Compensator</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modelling Language</td>
</tr>
</tbody>
</table>
10 Appendices

10.1 Appendix A, Process Map – Trip Management

10.2 Appendix B, Process Map – Upgrade due to Manufacturing Fault

10.3 Appendix C, Process Map – Change Control Board

10.4 Appendix D, Context Diagram – Remote information collection

10.5 Appendix E, Context Diagram – Remote upgrade
11 Bibliography


White, Stephen A. *Introduction to BPMN.* s.l.: IBM Corporation.


Appendix B
Appendix D, Context Diagram

Remote information collection from customer’s site

Company X
Aftersales / Coordinator / Service Engineer / Technical Specialist

System/function

Customer’s site

Protection devices

GWS
OWS/SER
MACH2
SMC
Appendix E, Context Diagram

Company X
Aftersales / Coordinator / Service Engineer / Technical Specialist

Remote upgrade of customer's site

System/function X

Customer's site

OWS / SER
- OWS applications
- SER databases

GWS

SMC

Protection devices

Firewall / Switch

MACH2 (main computer)

PS93X (communication board)

Circuit boards (separate SW and functions)

IEC-61850 DNP

CAN / eCAT