Requirement management

Barriers preventing the establishment of structured methods in software development

MARCUS KARLSSON
VICTOR LJUNG

Master of Science Thesis
Stockholm, Sweden 2014
Requirement management
Barriers preventing the establishment of structured methods in software development

Marcus Karlsson
Victor Ljung
Abstract

A major problem for companies within the software development industry is that their customers request more advanced technology and at the same time, there is a demand for products being developed in a faster pace. Meanwhile, there is a risk that the demands from the customers might change during the development phase, as a consequence of an industry that is characterized by continuous development of more advanced technology and a competitive setting. There is a great challenge for companies to cope with these changes and by not managing these correctly, severe economic consequences might occur.

In order to manage these situations, the importance of a requirement engineering process is emphasized by previous research. However, even though a process has been implemented throughout an organization, there are still factors that prevent it from being established over time. This study focuses on the barriers that prevent the use of a structured method in requirement management, by conducting a case study at Scania. The environmental context was software development for embedded systems.

Data collections in form of interviews were conducted at different departments at Scania to investigate existing barriers in the development process regarding requirements. The results from the respondents indicate problems with the requirement management from different perspectives. Aspects such as lack of time, understanding of the development chain as well as the frequent changes in design are emphasized as barriers for adapting to the existing requirement management process, leading to that the involvement of different user needs are disregarded.

Through these results, conclusions are drawn which will serve as a base for future recommendations to manage the requirement management process. These comprises the importance of identifying relevant stakeholders, of how new processes and tools are distributed among employees and the importance of continuous education of the newly implemented processes. This intend to highlight better ways of managing different user needs that hopefully, in the end, will lead to that more requirements are being considered at an earlier stage in the development.
### Sammanfattning

Ett stort problem för företag som utvecklar mjukvara är att kunden kräver allt mer av systemen samtidigt som de vill att utvecklingen ska gå fortare. På samma gång finns det en risk att användarnas krav på systemen ändras under utvecklingsfasen som en följd av att industribranschen karaktäriseras av ständig utveckling och hård konkurrens. Att hantera dessa ändringar är en utmaning och misslyckad kravhantering kan leda till stora ekonomiska konsekvenser för företaget.

För att hantera detta framhävs vikten av att ett företag använder en kravhanteringsprocess. Men även om en kravhanteringsprocess finns implementerad på företaget finns det flera faktorer som bidrar till att den inte hålls etablerad med tiden. Den här studien har därför undersökt vilka barriärer som kan förhindra användandet av en strukturerad metod i kravhanteringen genom en fallstudie på Scania. Kontexten på fallstudien har varit mjukvaruutveckling för inbyggda system.


Utifrån dessa resultat så förses rekommendationer för att hantera dessa barriärer så att flera användarbehov tillgododes i ett tidigt skede i utvecklingen och därmed sparar pengar i för företag. Dessa rekommendationer innefattar vikten av att identifiera relevanta användare, av hur man distribuerar nya processer och verktyg samt vikten av kontinuerlig utbildning och uppdatering av processer för medarbetarna.
ACKNOWLEDGEMENTS

We would first and foremost like to thank our industrial supervisor at Scania, Peter Sidén, for introducing us to this master thesis and inspiring us throughout the project. Our perspective on product development has definitely grown on many levels and without your guidance and continuous encouragement, this thesis would not have become what it is.

Many thanks go to our supervisor at KTH, Jens Hemphälä, whose contributions were invaluable. The discussions, the support and the questioning helped us greatly in forming the end result of this study.

Also, thanks to Carl Wadell who shared his expertise in the initial stage of the study by questioning our research approach and guiding us in defining our research question.

On a more personal note, we would like to thank the students who also chose to write their master thesis at the department of REV; Henrik Felixson, Joakim Bergius and Fredrik Hillnertz. The encouraging atmosphere that was created among us students was a contributing factor to that the spirit remained high throughout the project.

Additionally, we would like to thank all the employees at the department of REVM. You all were very open-hearted and welcoming which provided us with the right conditions at Scania, right from the beginning.

Our special thanks go to Magnus Selldén, who on numerous occasions used his professional expertise to provide valuable inputs within the area of interest, when our current knowledge was limited. The time you invested in our research was highly appreciated by us, thank you.

Lastly, we would also like to thank all the respondents who chose to participate in this study. The time and effort you contributed with, by sharing valuable experiences and ideas, enabled us to conduct this study.

Marcus Karlsson & Victor Ljung

Stockholm, 17/6-2014
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AER</td>
<td>Allocation element requirement</td>
</tr>
<tr>
<td>CFT</td>
<td>Cross-functional teams</td>
</tr>
<tr>
<td>COO</td>
<td>Coordinator</td>
</tr>
<tr>
<td>COO8</td>
<td>Coordinator version 8</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic control unit</td>
</tr>
<tr>
<td>FAD</td>
<td>Functional Architecture Description</td>
</tr>
<tr>
<td>GT</td>
<td>Grounded Theory</td>
</tr>
<tr>
<td>NFR</td>
<td>Non-functional requirements</td>
</tr>
<tr>
<td>RCI</td>
<td>Department of Driver Interface, Scania</td>
</tr>
<tr>
<td>REV</td>
<td>Department of Vehicle Control Systems, Scania</td>
</tr>
<tr>
<td>REVM</td>
<td>Department of Vehicle Management Control, Scania</td>
</tr>
<tr>
<td>UFR</td>
<td>User function requirement</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

1 INTRODUCTION ......................................................................................................................... 1
  1.1 BACKGROUND AND PROBLEM DESCRIPTION ................................................................. 1
  1.2 PROBLEM DEFINITION ........................................................................................................ 2
  1.3 DELIMITATIONS .................................................................................................................. 3
  1.4 CONTRIBUTIONS .................................................................................................................. 3
  1.5 OUTLINE ............................................................................................................................. 3

2 METHODOLOGY ...................................................................................................................... 4
  2.1 RESEARCH APPROACH ....................................................................................................... 4
  2.2 PRE-STUDY ......................................................................................................................... 5
  2.3 DATA COLLECTION ............................................................................................................. 6
  2.4 INFORMATION PROCESSING .............................................................................................. 7
  2.5 QUALITY OF THE STUDY ................................................................................................... 8

3 THE CONTEXT OF THE STUDY ............................................................................................ 10
  3.1 SCANIA .................................................................................................................................. 10
  3.2 REVEM ............................................................................................................................... 11

4 FRAME OF REFERENCE ......................................................................................................... 14
  4.1 USER NEEDS IN SOFTWARE DEVELOPMENT ................................................................. 14
  4.2 INTEGRATION OF STAKEHOLDER NEEDS ...................................................................... 15
  4.3 TYPES OF REQUIREMENTS .............................................................................................. 16
  4.4 REQUIREMENTS ENGINEERING PROCESS ...................................................................... 16
  4.5 CHALLENGES IN MANAGING A REQUIREMENT ENGINEERING PROCESS .................... 18

5 RESULTS .................................................................................................................................... 20
  5.1 RESULTS FROM THE PRE-STUDY .................................................................................. 20
  5.2 RESULTS FROM THE DATA COLLECTION ....................................................................... 22

6 ANALYSIS ............................................................................................................................... 28
  6.1 IDENTIFYING AND UNDERSTANDING OTHER STAKEHOLDER NEEDS .................... 28
  6.2 LACK OF TIME .................................................................................................................. 29
  6.3 REQUIREMENT CHANGES AND ADMINISTRATION ...................................................... 29

7 DISCUSSION AND CONCLUSIONS ....................................................................................... 30
  7.1 DISCUSSION ....................................................................................................................... 30
  7.2 CONCLUSIONS ................................................................................................................... 34

8 RECOMMENDATIONS AND FUTURE RESEARCH .................................................................. 35
  8.1 RECOMMENDATIONS ......................................................................................................... 35
  8.2 FUTURE RESEARCH .......................................................................................................... 36

9 REFERENCES ............................................................................................................................ 37

APPENDIX A: EXPLORATIVE INTERVIEWS GUIDELINE ............................................................... 1
APPENDIX B: INTERVIEW GUIDELINE ......................................................................................... III
1 INTRODUCTION

This chapter presents a brief introduction of the master thesis. Firstly described are the background, moving over to the problem definition, delimitations, contributions and the outline.

1.1 Background

Today, the dynamic and complex nature of software development is widely known to pose problems for organizations in their development efforts (Rehman, et al., 2013). This dynamic setting is created from an industry where the technology is moving fast and the users expect more and more from the available products on the market. As a consequence of the frequent change in demands, system requirements can be hard to define initially and tend to emerge during the development process. Acquiring and handling these requirements in a dynamic setting remains a challenge, leading to a risk of late changes in the design that will have significant economic consequences for the organization (Davey & Cope, 2008).

Requirements originate from user needs, which are the differences between the users’ goals and the present condition (Kujala, et al., 2001). These needs can be represented by both problems and possibilities that need to be transformed into requirements upon a system. However, representing a user need into a specified, technical requirement that is both understandable and verifiable by the user can be problematic. Studies show that incomplete requirements are a common reason for the failure of a project (Hull, et al., 2005). Moreover, user needs are widely recognized as an important aspect for a project’s success but challenges with measuring, documenting and validating exists (Glinz, 2007). These challenges are due to user needs being hard to define and their sometimes high level of abstraction. It is therefore vital to fully understand the user and the environmental context of the product in order to ensure the usefulness of the deliverable (Kujala, et al., 2001). As the requirements play a significant role in the project outcome, the need for an established way of capturing those remains equally important, disregarding the dynamic nature of software development.

Traditional development approaches such as waterfall models, where requirements often are considered frozen once the project starts and thereby becomes difficult to change at a later stage, are becoming less compatible with the dynamic nature of the software industry (Malik, et al., 2013). Agile approaches are more widely used in software development due to its compatibility with the software lifecycle that is characterized by frequent changes and updates in requirements (Malik, et al., 2013). Therefore, in software development, a requirement engineering process is commonly referred to in the management of requirements (Haron, et al., 2012). It is a framework developed to assist in the elicitation, documentation and verification of requirements from various stakeholders and aims to systematically iterate this procedure during the development process in order to keep them up to date (Rehman, et al., 2013). Having a requirement engineering process offers many benefits, where one important aspect is to create mutual understanding between the supplier and its customer to ensure that the demands of the involved partners correlates with one another (Broy, 1997). Furthermore, to implement a requirement engineering process successfully, developers must be able to use appropriate support methods necessary to manage the steps in the requirement engineering framework in order to make the requirements more tangible (Pandey, et al., 2010).

Although a widespread of models and methods has been developed to manage the requirement engineering process, making them generalizable in different contexts and establishing them in organizations in a structured manner is proven to be problematic (Kauppinen, et al., 2004). These
problems occur since developers focus primarily on the technical aspects and less on the user needs and changing this mindsets takes time. In accordance with this, Driessen & Hillebrand (2013) sees the ability to capture user needs as a capability which follows a learning curve that is developed over time. Thus, companies that work proactive with user needs are claimed to be more likely to consider a broader amount of user needs into their development (Driessen & Hillebrand, 2013). To embrace this capability, the organization needs to integrate stakeholder issue identification techniques, coordination mechanisms and prioritization principles, hence compile experience from integrating multiple stakeholders (Driessen & Hillebrand, 2013).

Additionally, after a clear, structured model for handling the requirement process have been implemented within an organization, there is still room for errors in the requirement management that is not directly connected to the outline of the developed process but rather how it is perceived and compatible with each employee. Most theory discusses the problem with implementing the process organizational-wide, such as the study by Kauppinen et al. (2004). Meanwhile, the theoretical framework covering the reasons why employees fail to keep an implemented requirement engineering method established over time in the organization is scarce and Kauppinen et al. (2004) mention this as an interesting area for future research to extend their study.

1.2 Problem definition

As the previous chapter indicated, research in the implementation phase of a new process in requirement management has been touched upon. In the study by Kauppinen et al. (2004), the factors that affected the implementation of a new process were investigated by studying organizations that were introducing and defining a new process of managing their requirements. In other words, implementation in that context meant introducing a change within the company environment, addressing the willingness to accept a change into the organization. Implementation in that sense is then mentioned to be the practice of activities that highlight the existence of the process as well as ensuring its practical use and that it satisfies the current needs in the relevant development projects. The implementation should result in the process being introduced and considered as approved to use (Kauppinen, et al., 2004).

As the results from Kauppinen et al. (2004) provided valuable insights when planning to introduce a new process into an organization, it would extend their perspective on the continuous improvement procedure of process development by following up on the impact of the new process as well as identify areas that can be problematic after it has been introduced in the organizational context, hence address factors preventing it from being established over time. A company might be willing to see the benefits and adopt the introduction of a new process at a certain stage, but circumstances might change in the future (Niazi, et al., 2005). Factors, such as compatibility with other, new processes and employee turnover, might affect the circumstances from the initial implementation stage to how it is compatible over time with each individual at the firm. Implementation refers to introducing and making its purpose clear enough to make it acceptable initially, but also to establish it by monitoring and maintaining its relevance to ensure compatibility over time. With this in mind, this study will address the barriers that might prevent employees to utilize a structured process in requirement management. Hence, the problem statement for this thesis will be the following:

What barriers are preventing the establishment of structured methods within requirement management?
1.3 **Delimitations**

Due to the scope of this project, focus will be on the barriers that prevent employees to use a structured method for requirement engineering in one case study, hence one specific context. The case study is performed at Scania and only internal stakeholders will be interviewed. This is due to the fact that those are the ones that will interpret and direct their development work according to the gathered requirements. Hence, external stakeholders such as drivers and suppliers are excluded. The thesis also focuses on the area of software development, in particular embedded systems. Due to the time frame of this study, the extent of which the impacts of these barriers have on the development is not included.

1.4 **Contributions**

This master thesis aims to contribute to the field of requirement engineering by investigating existing barriers that prevents employees from using structured methods to cope with requirements. The contribution will primarily indicate barriers, hence help companies put up strategies to overcome these and make their requirement engineering process more efficient.

1.5 **Outline**

The outline of the Master thesis is as following:

- Introduction
- The context
- Results
- Discussion & conclusions
- Methodology
- Frame of reference
- Analysis
- Recommendations & Future research
In this chapter, the methodology used in the project is described in order to present how the work process was executed. This is followed by a discussion of the validity and reliability of the chosen methods.

2.1 Research approach

The study started with a problem area, research within non-functional requirements (NFR) in software development, which was wide and fuzzy, the authors choose to have a qualitative approach with inspiration from grounded theory (GT). As opposed to quantitative approach, which is not appropriate in explaining motivation and psychological factors in depth (Dilanthish, et al., 2001), a qualitative approach is the preferable method to practice in order to obtain an understanding of how participants perceive their roles and tasks in an organization (Merriam, 1995). A GT approach means that the research problem would be generated through observations from real world situations instead of merely from abstract reasoning (Hoda, et al., 2011). However, a literature study was made simultaneously within the field of requirement engineering. The primary objective with this approach is to establish a research problem area that exists in the industry and simultaneously contribute to the academia.

Furthermore, the research process was conducted in three main phases with several steps, see Figure 1, which will be explained in this chapter.

Figure 1. The processes of the Master thesis.
2.2 Pre-study

The pre-study was primarily conducted in order to identify and clarify the existing challenges within the initial problem area. The entire phase is explorative and included three different methods which all contribute to the final delivery of this phase, the problem definition.

2.2.1 Literature study

An extensive literature study was performed in order to get a broad perspective in the research area of non-functional requirements. This was done by studying previous research, mainly in form of scientific articles and books, concerning non-functional requirements. This intends to enhance the authors’ understanding, by providing a solid base of information within the area. As the pre-study proceeded, the literature study also supposed to help define the research question and point out in which area the study would contribute with new knowledge. In addition, the literature study contributed with research methodology e.g. stakeholder analysis, interviews guidelines and methods for analysis.

The main data bases that have been used are Science direct, Scopus, Google scholar, KTHB primo. The following key words, all related to the chosen subject and industry environment, were used in the literature study: Non-functional requirements, non-functional requirements for embedded systems, embedded systems, managing non-functional requirements, frameworks for non-functional requirements, automotive industry, functional requirements, measuring non-functional requirements.

2.2.2 Explorative interviews

In order to grasp the problems concerning NFRs within software development from an industrial point of view, explorative interviews were conducted at different departments regarding software development at Scania. Hence, the aim with the explorative interviews was to capture issues directly related to the industry. The interviews were mainly conducted through open-ended questions regarding NFRs and problems regarding them. In addition to these questions, in order to understand their development process more closely where these NFRs occurred, the respondents were asked to illustrate which stakeholders that were involved in the process. This information was later then used as the base for the stakeholder analysis. The guideline used in the explorative interviews is attached in Appendix A.

2.2.3 Stakeholder analysis

Identifying relevant interviewees that are best informed in the context being researched is crucial for the study (Voss, et al., 2002). Therefore a stakeholder analysis was performed. The stakeholder analysis was based on eight different questions (The World Bank, 1996) and conducted with a brainstorming session, where stakeholders were identified and categorized. The mapping from the explorative study would complement this session with essential information about the stakeholders. The questions are as follows:

1. Who might be affected (positively or negatively) by the development concern to be addressed?
2. Who are the “voiceless” for whom special efforts may have to be made?
3. Who are the representatives of those likely to be affected?
4. Who is responsible for what is intended?
5. Who is likely to mobilize for or against what is intended?
6. Who can make what is intended more effective through their participation or less effective through non-participation or outright opposition?
7. Who can contribute financial and technical resources?
8. Whose behaviour has to change for the effort to succeed?

Furthermore, to reduce the risk of missing out any important stakeholder, interviews were performed with stakeholders to complement a brainstorm session (Reed, et al., 2009).

2.2.4 Problem definition

From the explorative interviews, the literature study and the stakeholder analysis, a problem definition could be generated. However, since several problem areas would emerge, the criteria for the chosen problem area would be that it affects several participants in the explorative interviews as well as contributes with something new within the research area.

2.3 Data collection

After formulating the problem definition and identifying relevant stakeholders, a qualitative data collection session was performed. This was mainly done through a new literature study and semi-structured interviews.

2.3.1 Literature study

Having the problem area defined creates a need to complement the literature with theory that is more directed towards this area. This literature study would also help formulating interview questions that would be relevant to ask. This literature study used the same data bases as the literature study in the pre-study phase. The following key words were used, where focus was put on incentives in including requirement in development efforts: Non-functional requirements, user needs, incentives in requirement engineering, requirement engineering process, functional requirements, implementation problems, integrating stakeholder needs.

2.3.2 Interviews

The chosen method to obtain the qualitative data was semi-structured interviews since it is considered as one of the best techniques to learn about people’s behaviours and choices (Raworth, 2012). Semi-structured interviews are feasible with explorative studies, since they are conducted in a conversational manner, giving the participants the opportunity to elaborate on matters they feel are important (Longhurst, 2003). Hence its compatibility with this study, as the purpose with the interviews was to fully understand the challenges employees perceive in the development.

Furthermore, extracting information about a relatively unexplored and unfamiliar area for the respondents could be a challenge. This requires the interview guide to be designed carefully and correct so it fits the research topic and is interpreted and understood the same by all respondents. Guidelines for setting up and conducting a semi-structured interview is presented by Raworth (2012) and worked as a base for the interview guide of this study, see Appendix B.

Information about the respondents in this study is presented in Table 1:
<table>
<thead>
<tr>
<th>Title</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>System owner</td>
<td>REVM</td>
</tr>
<tr>
<td>System owner</td>
<td>REVM</td>
</tr>
<tr>
<td>Function owner</td>
<td>REVM, REVE, NEC</td>
</tr>
<tr>
<td>Function owner</td>
<td>REVM</td>
</tr>
<tr>
<td>System owner and function owner</td>
<td>RCIC</td>
</tr>
<tr>
<td>System owner</td>
<td>RCII</td>
</tr>
<tr>
<td>System owner</td>
<td>RCIC</td>
</tr>
<tr>
<td>Product coordinator</td>
<td>RTPS</td>
</tr>
<tr>
<td>System architect</td>
<td>RESD</td>
</tr>
<tr>
<td>Senior engineer – Embedded systems test strategy</td>
<td>REST</td>
</tr>
<tr>
<td>Expert engineer – Service support solutions</td>
<td>YSPX</td>
</tr>
<tr>
<td>Senior engineer – Service support solutions</td>
<td>YSPX</td>
</tr>
<tr>
<td>Development engineer</td>
<td>YSPX</td>
</tr>
<tr>
<td>Repair methods engineer</td>
<td>YSRE</td>
</tr>
</tbody>
</table>

Each interview took approximately one hour to perform. The interviews were carried out by the two authors, one with a leading interviewing role and one with a leading data collection role. To fully cover the interviewees’ answers, all interviews were recorded and then transcribed.

### 2.4 Information processing

The last phase included processing of the data that was collected in the second phase. This comprised of a literature study, an analysis, discussions, conclusions and recommendations.
2.4.1 Literature study

This literature study aims to investigate whether the results of the interview correlates with the existing theory. Hence, narrow down the existing literature study to what would be most relevant to analyze and complement if necessary. The complementation was mainly dependent on if the results would deviate or shed light on a specific aspect. Therefore, the following key words that were used were related to the answers from the interviews: Manage user needs, incentives in requirement engineering, establishing requirement engineering process, implementation problems within requirement engineering, problems with integrating stakeholder needs, improve requirement engineering process.

2.4.2 Analysis method

Since most of this research is based on interviews, analyzing the transcripts needs to be well worked through. First of all, vast amount of data is gathered from the interviews. In order to process this information efficiently, the fourteen stages proposed by Burnard (1991) was used, see below.

1. Take notes and write memos throughout the project.
2. Transcripts should be read through and notes made.
3. Transcripts should be read through again.
4. The list of categories is grouped together under headings.
5. The list of categories and sub-headings is worked through.
6. Colleagues review the categories.
7. Transcripts are re-read.
8. Each transcript is worked through with the list of categories and headings and coded.
9. Each coded section is cut out from the transcript and put together.
10. The cut out sections are pasted onto sheets with appropriate headings.
11. Selected respondents are asked to check the appropriateness of the category system.
12. All of the sections are filed together for direct reference.
13. Writing up, linking the examples in each section together. Takes one section at a time.
14. The last stage is about whether to link the data to the literature or not.

Secondly, one has to consider whether the interviewees can be compared or not. Hence, to see if the respondents shared the same perspective on common themes.

Moreover, data that is not considered to be categorical or contributing to the general understanding of the field may be excluded (Pope, et al., 2000).

2.5 Quality of the study

The quality of the study is herein discussed in terms of validity, reliability and generalizability.

2.5.1 Validity

In qualitative research, validity refers to which extent the findings are true and accurate (Guion, et al., 2002). This includes the aspects of whether the research findings reflect the situation accurately and if they are supported by the evidence. Although qualitative studies are hard to
validate, many researchers agree that some sort of qualitative measure is desired (Golafshani, 2003).

In order to achieve higher quality, the case study will include two different departments at Scania working with software development. Having two different departments with customized processes generates a broader perspective to this matter. Moreover, fourteen people at these departments were interviewed, all with a minimum experience of one project, to increase the credibility and obtain a wide set of perspectives and insights. The interviewees had the opportunity to validate their insights and how they affect other stakeholders afterwards.

2.5.2 Reliability

The reliability is the extent to which results of a study can be reproduced and repeated (Golafshani, 2003). However, in qualitative research, reproducibility and repeatability is difficult since circumstances change. One reason for this is, when humans are involved, their behaviour is never static. Changes in the world environment affect people’s mind-sets according to their surroundings (Merriam, 1995). Since there are two authors of this thesis and both are participating in the interviews, no conclusions will be made until consensus has been reached which is suggested, by Voss et al. (2002), to enhance the reliability.

2.5.3 Generalizability

To draw general conclusions about behaviours and events from a single case study is somewhat troublesome (Gomm, et al., 2000). To increase generalizability, the case of the research should be as typical as possible (Gomm, et al., 2000). However, in this case study, due to its specific process, industry conditions and products, generalizability would be difficult. Besides this, the development that has been considered in this study is dependent on the modularity of the company’s products. Companies which do not have this type of modularity approach to its products might face other problems. It is important to consider this when conclusions are to be drawn and when arguing for whether these are generalizable or case specific.
3 THE CONTEXT OF THE STUDY

Herein, the context in which this study has been conducted is described. This includes the company, the processes and how requirements are being handled.

<table>
<thead>
<tr>
<th>Introduction</th>
<th>The context</th>
<th>Results</th>
<th>Discussion &amp; conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methodology</td>
<td>Frame of reference</td>
<td>Analysis</td>
<td>Recommendations &amp; Future research</td>
</tr>
</tbody>
</table>

3.1 Scania

The case study was performed at Scania, one of the world’s leading manufacturers of heavy trucks and buses. The company operates worldwide and has more than 38,500 employees whereas around 3,500 work within research and development, mainly situated in Södertälje, Sweden. Development cycles at Scania take years and involve several departments before released on the market. Some of their products are illustrated in Figure 2.

![Figure 2. Scania trucks.](image)

Scania identified problems regarding their current way of managing requirements, where some stakeholders’ needs might not get considered. Methods for handling the requirements existed throughout the organization, yet how it was followed and perceived by different individuals at the firm was not known, leading to question the efficiency of the implemented process. Therefore, the purpose behind this thesis is to evaluate Scania’s requirement process. The study will be performed in the context of software development, with a case study conducted at the embedded systems department at Scania. The case study intends to provide insights in how user needs are handled today and why developers might disregard the involvement of certain user needs in their development.
**3.2 REVM**

The master thesis was mainly conducted at the department of REVM, which develops and maintains one of the vehicles electrical control unit (ECU), the Coordinator (COO). REVM is a part of the section REV, involving different departments responsible for the control systems of the vehicle. This means that they work with an embedded system and therefore primarily software development. How the coordinator is related to the other parts of the vehicle is illustrated in Figure 3.

![Figure 3. Communication of the COO.](image)

The ones who work at REVM are system owners, function owners and function developers. System owners are responsible for a system, e.g. an ECU such as the coordinator, while function owners are responsible for functions, e.g. the cruise control. This means that the function owner responsibilities can extend to other systems as well as illustrated in Figure 4.

![Figure 4. Domains for Function/System owners.](image)

The responsibility of function developers is e.g. to implement the function in different software’s and systems.
3.2.1 Parameters

To narrow the research down and to create a more focused development area, the case study concentrated on the configuration development of their products. This primarily means how they set parameters that will steer how their functions will work. Without describing parameters in depth, developers mainly design parameters in order to allow different configurations and options to their functions and systems. How these parameters are designed and set will affect several stakeholders, including service market, integration test, drivers and production. It is therefore crucial for the developers to have all stakeholder needs clear and aim to fulfil their needs to the greatest extent.

3.2.2 Processes

The product development process followed a release process that was synchronized with the whole company, including software and hardware development. This was necessary because of the complexity of the product and the involvement of many departments. Further, the release process was based on the waterfall principle with gates where different managers and stakeholders give their opinion to the work and choose whether the development can proceed or not. Scania has chosen to divide the development processes into three types; process in concept development, process in product development and process in product follow-up. During the concept development, customer demands were turned into technical solutions, risks were evaluated and preparations for product development were made. When this was approved, the project proceeded into the process of product development. This process starts with a concept and ends with a product ready for production. During this time, several cross-functional milestones should have been passed including risk and safety analysis, software freeze, virtual and physical testing and product verification. The third process with product follow-up would be actuated if the released product required updates or maintenance.

Furthermore, REV has made a customized product development process in order to fit their specific needs and conditions. This process is linked with the release process through milestones but the content in between differs from other departments. The V-model below, see Figure 5, highlights the most important steps in their process.

![Figure 5. V-model for REV’s development process.](image-url)
The personnel at REV often work close together within their department and with the other departments located in the same building. The ones they work close with include system testers and system architectures for the coordinator.

3.2.3 Requirements

Requirements for new function development are primarily being handled through documents; User function requirements (UFR’s) and Allocation element requirements (AER’s). How they fit together is described by a Function architecture description (FAD).

UFR

UFR’s are used to describe the user function and which requirements it should fulfill, regardless how the implementation should look like. The purpose is to create an understanding for how the function works and this document will work as the base for other departments at Scania who should work with the function. This document generally includes three main areas; introduction, requirements and a hazardous analysis.

The UFR should be used for function and integration test, design, for hazard analysis, for service market. It should also work as a source for product acquaintance.

AER

AER’s includes requirements and the description of allocation elements (AE). An allocation element is a logical component which is realized with a physical component in the ECU of the vehicle.

The AER should control how the implementation in the ECU should look like, describe what should be tested at module and system level, for hazard analysis, what service market needs and for dependency analysis.

FAD

The FAD is primarily made to document how a function is realized with an AE. It also states how the development of a function progressed. The FAD generally describes the correlation between the UF and the AE.

It is used for function and system testing, design, hazard analysis, service market and for a general dependency analysis for the system.
4 FRAME OF REFERENCE

This chapter aims to present the relevant frame of reference to this study. The reader is intended to get insights in how the literature has approached these areas and how this relates to this study.

4.1 User needs in software development

User needs are vital to consider in software development as they are known to cost companies a significant amount of money if managed poorly (Browne & Rogich, 2001). According to Urquhart (1999), 75% of all error removal costs are related to correcting mistakes made during the elicitation of requirements at the design phase. Another important aspect to consider is that usability and user involvement often are not prioritized when there is time pressure, and in software development, high time pressure is often the case (Cysneiros & Yu, 2004). This is a consequence of customers expecting their products to be delivered in a faster pace.

Mentioning user needs, the intended needs are those needs coming from internal and external stakeholders affected by the developed product. A stakeholder group can be defined as “any group or individual who can affect or is affected by the achievement of the organization’s objectives” (Freeman, 1984, p. 46), essentially meaning that they all have stakes to either gain or lose, based on the outcome of a project. Therefore, covering and understanding all stakeholders is important in order to retrieve sufficient requirements.

According to Pikkarainen et al. (2008), there are five different key stakeholders in software development, see Table 2.
<table>
<thead>
<tr>
<th>The Software Development Team</th>
<th>Responsible for the development of the software</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Management</td>
<td>Defines the strategy for the organization</td>
</tr>
<tr>
<td>The Enterprise Staff</td>
<td>Other employers at the company who either affect or maintain the product, for example groups who develop products with connecting interfaces</td>
</tr>
<tr>
<td>The External Customers</td>
<td>Customers, Users, Sales group</td>
</tr>
<tr>
<td>The Support Group</td>
<td>Supports the software, for example quality ensuring, testing, facilitates etc.</td>
</tr>
</tbody>
</table>

Additionally, Pikkarainen et al. (2008) highlights that it is common for the needs of different stakeholders to interfere with each other, causing a need to handle trade-offs and understand the consequences. Therefore, how stakeholders are integrated in the development process is of importance when considering the output from the project.

### 4.2 Integration of stakeholder needs

Communication remains a primary activity to transfer knowledge regarding requirements upon a system (Pikkarainen, et al., 2008). Problems meeting these requirements involves that different stakeholders communicate from different perspectives, valuing different features. Therefore, communication is a vital aspect in development projects. Information exchange within an organization involves both time and cost and finding the right balance in how frequent this occurs is important. Infrequent exchange would increase the chance of design flaws, while too frequently would also be a negative factor, as costs would increase. (Lin, et al., 2010)

Zhao et al. (2011, p. 19) refer to internal integration as “the degree to which a firm can structure its organizational practices, procedures and behaviors into collaborative, synchronized and manageable processes in order to fulfill customer requirements”, essentially meaning that organizations should not consider different functions or stakeholders at the firm as independent, but rather encourage information sharing between them to increase the performance of the outcome.

Different approaches in the interaction between two stakeholders have been studied, as in the study by Fricker et al. (2008) where communication regarding requirements is preferred to be goal-oriented. However, as the communication has to go through several layers of stakeholders within the development chain, the situation becomes more complex and the risk of misinterpretations are more common (Damien, 2007). Damien (2007) continue by emphasizing the need of a clear organizational structure with communicating responsibilities for distributed projects to help improve the communication between several layers of stakeholders.
Today, cross-functional teams (CFTs) are commonly used by new product development firms in order to increase coordination and integration capabilities as well as to improve organizational boundaries (Feng, et al., 2009). This will, in turn, increase the understanding of the different customer needs, which will be directly linked to the success rate of the project (Hirunyawipada, et al., 2010). Hirunyawipada et al. (2010) continues by arguing that applying CFTs to enhance the communication is not the main factor for success, but rather the team’s capability of transforming shared information into usable knowledge. This is because developers often see the system in terms of its data structure and procedures while end users emphasize general behaviour, functionality and applications (Al-Rawas & Easterbrook, 1996). Therefore, the classification of different types of requirements is important to understand.

### 4.3 Types of requirements

Two different types of requirements exist within software development, functional requirements and non-functional requirements (NFR) (Pandey, et al., 2010). While definitions regarding functional requirements have been emphasized, describing what a system does (Glinz, 2007) (Ackermann, et al., 2008), less focus has been put on the non-functional requirements, which also inflict upon the outcome of a project (Glinz, 2007).

According to Saadatmand (2012), there is no consensus in the field of requirements engineering of how to define NFRs, but continues by suggesting that one way to describe a NFR is as a requirement for how a system should act. Ackermann et al. (2008) refers to the topic in a similar way, stating that system-related and structural properties, for example how well software perform, are usually named non-functional requirements. Another way to define it, according to Martin Glinz (2007), is that a NFR is an attribute of or a constraint on a system. Here, an attribute is suggested to be a performance requirement or a specific quality requirement and constraints described as off-limits to design trade-offs. Broy (1997) supports this, addressing non-functional requirements as something that takes the system constraints into account and thereby will affect the functional requirements, and puts emphasis on that it is beneficial to state the relevant non-functional requirements while documenting functional requirements.

The relation between functional and non-functional requirement can be hard to handle, as illustrated in the study conducted by Cao & Ramesh (2008). Out of 16 participating companies, 14 of them did not pre-define their requirements but rather let them emerge during the projects. The reason behind this claimed to be because of high requirements volatility, lack of relevant knowledge within that area and that customers had a hard time expressing their requirements before they actually saw the outcome. A consequence of this was mentioned to be the neglecting of non-functional requirements, as focus is instead put on getting the core functionality correct. Aspects such as maintainability and usability were disregarded during the development, leading to late changes in the design. This urges the need to manage the requirements in a structured and predictable way.

### 4.4 Requirements engineering process

Requirements engineering is described to be an important part of software development to ensure quality and meet customer needs, due to the rapid changes in that market environment (Rehman, et al., 2013). The definition of requirement engineering, according to Zave (1997, p. 315), is that: “Requirements engineering is the branch of software engineering concerned with the real-world goals for functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.” Although this definition is old, it is still used and
relevant in today’s research (Rehman, et al., 2013) (Khan, et al., 2010). In other words, one main purpose of practicing requirements engineering is to get assistance in knowing what to develop before the actual developing work starts in order to prevent costly changes (Haron, et al., 2012).

The framework of the requirement engineering process is commonly classified into different phases (Pandey, et al., 2010) (Bhawna, et al., 2012). These phases are the following:

- Requirement elicitation
- Documentation of requirements
- Validation and verification of requirement
- Requirement management

These phases are considered to be iterative and a systematic approach to the requirement engineering process, suggested by Pandey, et al. (2010), is presented in Figure 6.

![Figure 6. An iterative approach to Requirements Engineering. (Pandey, et al., 2010)](image)
4.5 Challenges in managing a requirement engineering process

The aim of establishing an efficient requirement engineering process is to ensure that the output from the developers correlates with the desires of the involved stakeholders and the performance of the product is measured by the degree of acceptance of the latter (Fricker, et al., 2008). However, going from stating the project’s scope to a fully developed product is a challenging transition, where one example of this is if the requirements changes over time as the project progresses (Rehman, et al., 2013). The reasons behind this might be due to different circumstances; e.g. the customers might change their preferences or due to new regulations by the government. Davey & Cope (2008) suggest a dynamic approach to this process to cope with changes in opportunities and constraints, by avoiding looking at the requirements as a stable set. According to studies done by Lloyd et al. (2002), actively involving stakeholders in the process activities is desirable to keep the requirements up-to-date. Enhancing informal communication between stakeholders is also mentioned to make the requirements engineering process more effective.

Furthermore, in order to develop sufficient software architecture, eliciting and handling requirements throughout the process remains a key factor (Doerr, et al., 2005). However, eliciting these requirements, especially NFRs, is stated to be difficult (Doerr, et al., 2005) (Borg, et al., 2003) (Cysneiros, et al., 2001). One reason for this, according to Cysneiros et al. (2001), is that the stakeholders are not used to deal with NFRs, hence lack the ability to identify these essential requirements. There are also issues with stakeholders not understanding what they want, their low participation in reviews and that they insist on new requirements late in the process (Bhawna, et al., 2012). This urges the need for methods to manage the elicitation. The framework for the requirement engineering process described in Figure 6 is just an outline of the critical steps in the process flow. Research argue that the requirement engineering process has to be tailor-made, with a combination of techniques in each step of the requirement engineering process, according to the nature of the organizations’ environment (Rehman, et al., 2013). For example, observations and interviews might be effective techniques in the elicitation step (Pandey, et al., 2010).

Kauppinen et al. (2004) contributes to this field of knowledge by proposing critical factors that will affect the implementation of the requirement engineering process, organizational-wide. They define the process improvement procedure for an organization that wants to develop a requirement engineering process as following, see Figure 7.

![Figure 7. An approach to process improvement with focus on requirement engineering. (Kauppinen, et al., 2004)](image)

Critical success factors for the implementation activity of a new requirement engineering process are highlighted, where it is mentioned that a challenge with process improvement in software
environment is people’s resistance to change. Furthermore, Kauppinen et al. (2004) found that a change had to be introduced at an individual level at first, then proceed to a project level and lastly to an organizational level. One condition for an employee to practice a new method is mentioned to be when the results of it becomes transparent, creating a motivation factor to apply the same procedure in the future. A solution to this is mentioned to be training, educating employees why requirement engineering is important, how it performed and how it is related to the overall process of the organization. Additionally, the result of a method is mentioned to be dependent of more factors. When the results occurs is highlighted as relevant, since long term goals can seem too distant for an individual to feel motivated to reach. At the same time, whom the result affect is also important, meaning that not only the managers should benefit from it, but everyone contributing in the work effort.

Altogether, these factors are highlighted as human factors, involving motivation, commitment and enthusiasm and will play an important role in the implementation phase. Gaining a deeper insight of what role these human factors have is mentioned by Kauppinen et al. (2004) as an interesting area for future research. Based on their model of process improvement procedure, the next step, the monitoring of new requirement engineering processes, will be of interest in this study. This will, in contrast to Kauppinen et al. (2004) focus on the introduction and implementation phase, provide insights of what success factors that exist, to keep an implemented requirement engineering method established over time.
The results of the interviews at Scania are herein presented. This information was gathered from interviews with persons from development departments, system and integration test, marketing, sales, aftermarket, diagnostic architectures and tools, production and product coordination. Further, the results are categorized mainly with focus on stakeholder involvement, how knowledge is shared and problems that occur when usability needs from other stakeholders are neglected. The quotes that are being used from the interviews are freely translated from Swedish to English.

5.1 Results from the pre-study

As the study started out explorative in the area of requirement management, with a problem statement yet to be defined, focus initially was put on understanding Scania, their processes and what challenges that existed today at the organization regarding the subject.

5.1.1 Identification of stakeholders in the development chain

As the requirement upon a system could be related and affected by many stakeholders within the organization, it was of importance to fully understand the generic development chain within the departments working with embedded systems. However, as the chain was long and complex, many employees could only relate to the stakeholders in close contact with themselves and faced a hard time expressing every stakeholder in the chain. This meant that a definition of the internal development chain at Scania would not only be beneficial for the authors to conduct this study, but also serve its value for each employee so that they easier could relate to requirements of others. Therefore, one milestone with this study was to present a clear overview of this process.

In order to gain insights about their development processes to further identify relevant problem areas, five explorative interviews were conducted with persons at different departments at Scania. They shared their insights and experiences regarding the different responsibility areas throughout the development chain. Altogether, the results from these interviews were analysed and compiled and by complementing these with internal documents of their processes, a stakeholder map of the different stages could be created, see Figure 8.
As the study aimed to focus on the internal aspects of requirement management, the stakeholder analysis was limited to departments internally on Scania. Therefore, truck drivers, mechanics, logistic companies and suppliers were neglected in this study. The identified stakeholders and their contributions to the development are presented in Table 3.

Figure 8. Map of the stakeholders involved in the case study.
Table 3. The identified stakeholders and their main input to the development.

<table>
<thead>
<tr>
<th>Stakeholders</th>
<th>Input to development</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
<td>New customer needs</td>
<td>N/A</td>
</tr>
<tr>
<td>Sales</td>
<td>How functions relate to other functions</td>
<td>Rarely</td>
</tr>
<tr>
<td>System test</td>
<td>Test report</td>
<td>Once each iteration</td>
</tr>
<tr>
<td>Function owner</td>
<td>Responsible of a function</td>
<td>Frequently</td>
</tr>
<tr>
<td>Function developer</td>
<td>Develops specific parts of the function</td>
<td>Frequently</td>
</tr>
<tr>
<td>System owner</td>
<td>Responsible of a system</td>
<td>Frequently</td>
</tr>
<tr>
<td>Product coordinator</td>
<td>Helps out with new variants of FPC’s</td>
<td>Mainly early in the project</td>
</tr>
<tr>
<td>PSM-team</td>
<td>Facilitates tools for parameter setting and diagnostics</td>
<td>Once each iteration</td>
</tr>
<tr>
<td>Integration test</td>
<td>Test report for the whole vehicle</td>
<td>Once each iteration</td>
</tr>
<tr>
<td>Service market</td>
<td>How the mechanics will use the new function</td>
<td>Cross-functional milestones during the iterations</td>
</tr>
<tr>
<td>Production</td>
<td>Production requirements</td>
<td>Cross-functional milestones during the iterations</td>
</tr>
</tbody>
</table>

5.1.2 Problem definition

The problem definition was systematically developed in parallel with the initial literature study, the explorative interviews and the definition of the development chain. Compiling the results from these three areas enabled the emerging problem definition. There was a clear indication that the knowledge regarding requirement management existed within the organization, however it was poorly distributed among the employees. This resulted in a problem definition regarding the barriers that prevents employees to follow a structured process that was implemented by Scania.

5.2 Results from the data collection

The results were based mainly on the 14 interviews conducted at Scania and complemented with mail answers and discussions with sales, production, mechanics and technical advisors at Scania. The results obtained from the interviews revealed concerns regarding the current requirement management in software development. One problem area was recognized to be more emphasized by a majority of the interviewees; when developers implement their software before establishing requirements in a documented matter.
Moreover, different reasons to and aspects of this problem occurred, much related to experience and the professional role of the subject. Therefore the data from the interviews will be categorized into four different groups based on results.

5.2.1 Lack of time

The first aspect that caused developers to implement before setting up requirements was lack of time. This was also the most common reason that came up during the interviews. The respondents generally implied that it was more important to prioritize having the product finished in time and afterwards documenting how it went, which some interviewees mentioned as:

"Time, that is probably the problem. Putting effort into understanding other stakeholders’ needs just feels like a waste of time unless you have bumped into problems”

System owner B

"Insufficient understanding for what’s in the technical descriptions. There are too many different processes and deliveries, barely enough time to fill in and make everything. There is also barely no following up on what that has actually been done."

Function owner A

The problem with this was that many respondents were aware of the risks with neglecting requirements from the start and expressed a desire to work more proactive, yet this problem related to working reactively appeared to be common. As one person described how they work:

"Very reactive. Would have been good if you had a review internally, unfortunately you often experience time pressure and assume that this will do and goes on. If you look at hours spent in a project it is better to do right from the start. Even if it feels like, how can I save time if I spend more time on it?"

System owner B

Another problem that is emphasized is people’s incentives, where some felt that the time put into documentation is not worth it since, most of the time, there is barely no one that will read it or because their other commitments at work would suffer. This also led to prioritizing functional requirements, because of their direct relation to the product being useful, instead of non-functional requirements, which affects how the product that is being used is perceived.
“I think we put in an extreme amount of time in vain to complicate the specification structure internally. A huge amount of documents are written which very few reviews. It is often the case that we lack time for documentation or setting requirements before we develop.”

System owner A

“It takes too long time to get the requirement specification reviewed. It can take up to one month before you receive feedback. You do not have time with that so you start developing.”

System owner A

“It will be hard to accomplish anything if you have to consider all involved, gets less efficient.”

Function owner A

Example

There have been examples of when time pressure has been a problem at Scania. In one case, the function owner had taken over the responsibility for the function late in the project and he perceived that everything was poorly managed. There was also a lack of documentation and motivations from the predecessor why critical design decisions had been made. Therefore, to finish the project in time, focus was put on functionality whereas non-functional requirements such as usability of other stakeholders were done sparsely.

5.2.2 Understanding of other stakeholders needs

Another aspect that appeared was the understanding of other stakeholder needs and clear and understandable requirements. This was related to lack of documentation and lack of insight in other stakeholders’ role in the process and their tools. A common answer from the interviews was that it exists a poor understanding of how the whole process chain worked and which stakeholders that were included. The respondents described Scania’s process chain as very complex and that the projects often had different circumstances. This complexity often led to high emphasize on the individual’s own work and direct deliverables and how this affected the end customer.

“Good insight about the ones who are implementing and the one that should test your function, but about the other groups involved, the awareness is lower. So everything that concerns production, service market, repairing etc. is very often neglected I believe.”

System owner B

“I think the holistic view is still quite mediocre, seeing it from my perspective as last stakeholder in the chain. As soon as you have finished your own tasks you move on. It is focus on deliveries and not on responsibility over time.”

Service market A
Not knowing which stakeholders that would receive the deliverables made it harder to understand the importance of documentation and including all relevant requirements. A majority of the interviewees argued that it was easier to concentrate on the own deliverables and neglect responsibility for the outcome of the whole chain. People often strive to fulfil the functional requirements in time, but how these relate to NFRs such as usability and documentation is often disregarded.

Lacking a standard method to follow makes the work directly dependent on the individual carrying out the tasks. As a consequence, the way of work goes from structured to being subjectively chosen. Thus, inexperienced developers might miss out important aspects which mainly are built on experience such as prioritization. The respondents that have been stationed at other departments meant that they could easier identify and relate to other stakeholder’s needs. Prioritization and seeing the whole picture is emphasized in the citations below:

"It is very easy to take the requirements from production seriously. It is easier to diminish the requirements from the service market."

Repair method engineer A

“We have had to put enormous efforts on the system owners and function owner to make them see the whole picture. There are a lot with the right intention but few who actually do it.”

Service market C

Furthermore, problems with having requirements and requirement specifications differing in structure and clarity is pointed out as a hindrance for others to perform their work in a smooth and easy way. Some mentioned that requirements only were made for the testers so they could verify the function. This led to that the requirements would only be written with the testers in mind but no other stakeholders. One interviewee also pointed out that writing requirement was just a necessary mean, it was much more motivating to implement and test the function in practice.

Example

Many stressed the importance of having a holistic view over the product chain. One case mentioned by several respondents as a pitfall was when a new function was developed and where usability had been disregarded, in this particular case the alarm system. The requirements from the start enabled the driver to adjust the function in the truck in order to fulfil the specific regulations a country could have. However, the developers enabled all parameters which could be relevant as adjustable, which meant a lot. This made it very hard for the driver to know which ones to adjust and how. It also took a huge amount of time to conduct. The dependencies between different parameters were also hard to understand and took time. In the end, the driver had to go to a work shop to get help adjusting these parameters. However, the mechanics also found this difficult and time consuming but they could do it. This was an example of when the functional requirements was met, where the driver could change the function depending on the country he was driving in and its regulations, but the user friendliness was poor. To address this, one respondent mentioned:
“It is important for the developers to have the end users in mind when they systemize parameters. By who and how are the parameters going to be used? It is easy to think when you develop a function/system that some things might be good to be able to do, even though you lack the ability to see how and when it will be done. Where should it be physically done? You should, in general, always have the use cases clear for example: this will be adjusted by the mechanic at this moment. Having clear descriptions makes it significantly easier for us.”

Service market C

5.2.3 Requirement changes and administration

Another identified problem area with implementing before setting requirements is that the ability to deal with change and administration became worse. The developer often had the purposes tacit and only documented the end results. The problem that occurred was when someone else would take over an ongoing project or if something old needed updates. Since the documentation was lacking a lot of information and motivations of different design decisions, developers unfamiliar with the function felt insecure about changing things, because it was hard to see how the change would affect the system or other functions.

“Above all will the requirements get very specific but still miss out the question why it is set to that specific value. Hard to understand why the requirements get 1.02 m/s, where did that come from? Then it is often something that has been tested and tuned into that value. If you generally would describe a cruise control on a user-function level, then you would be able to go back and see what we had requested and break that down and see what we should do.”

System owner D

However, some of the respondents implied that it is not the initial requirements that is important but rather how you relate to requirement and changes during the project. Many pointed out the difficulties with setting fully defined requirements from the start and argued that it was better to motivate changes along the way. This would help to navigate the project in the right direction.

“IT feels unnecessary to force out requirements which later will be obsolete or unverifiable. It is better to motivate how and why with descriptions.”

System owner E

“It sometimes feels like the results is not that linked to the requirements you might write, rather how you handle changes during the project.”

System owner A
Even though requirements were documented, some stressed that meeting the person that would receive and try to interpret the requirements face-to-face would be necessary to avoid uncertainties.

Example
Before the COO8 was released there was great effort put into updating all specifications from an old way of documenting to the current AER template. During this updating, a lot of uncertain motivations to design decision were found. It was hard to make changes without knowing how this would affect the system, leading to that the requirements remained outdated. When working with the COO8 now, the developers at least try to question why they make certain decisions. It was easier to make a change in a new system, because of frequent testing. Then the changes could be verified and validated easily, to see if it worked or not. However, old systems or functions that was not in an active development project but still out in the market was harder and more uncertain to change.

5.2.4 Reason why developers specify requirements initially

Some reasons why developers would implement before setting requirements emerged due to costs associated with late changes. This was more present at the department of RCI where systems that were developed by external suppliers existed. RCI tried to specify as many detailed specifications as possible early on. If a need to change requirements later on in the project would occur, then it would often cost a significant amount of money. This specification was often discussed and revised several times between the supplier and the developer in order to gain a common understanding of the requirements.

"Changes costs money. If you care about the exact end results then you need to write it down."

System owner B
In this chapter, results are compared and analysed in relation to the literature found relevant. Some of the result lacks a counterpart in literature which indicates that these aspects have not been studied before. The areas that are being analysed are mainly mapping and understanding other stakeholder needs, the aspect lack of time and problems that occur with volatile requirements.

6.1 Identifying and understanding other stakeholder needs

Having a clear organizational structure is needed; otherwise misinterpretation could be made, particularly when several layers of stakeholders are involved (Damien, 2007). However, the case study implied that the organizational structure was complicated and that the insight in the whole chain was deficient. It was further described as a complex chain which could differ depending on the projects. As a result, people tended to put most emphasize on their own work and deliverables and how this affected the end customer.

Further, the developers in the case study lacked methods and tools to identify and map which stakeholders that was involved. Having a stakeholder template to start with, as Pikkarainen et al. (2008) illustrates, would help identifying stakeholders and their needs. Moreover, Pikkarainen et al. (2008) argues that different stakeholders’ needs often interfere with each other, making trade-offs important early in the process. Lack of stakeholder identification, hence stakeholder needs, might cause late requirements to surface and trade-offs harder to make. Some of the respondents of this study mentioned that trade-offs was rarely the problem but rather which stakeholders that was prioritized. For instance, requirements from production struck directly if they were not fulfilled, however requirements from service market could take years before anyone noticed any faults.

A majority of the respondents found that lack of documentation and poor insight in others roles and tools led to inadequate understanding of their requirements. This is something that can be addressed with learning-by-doing, which Kauppinen et al. (2004) suggests as an effective way of practicing implementation. The service market department invited developers to get familiar with their tools by having workshops. This was appreciated and experienced as a successful approach. However, far from every developer has done this with service market and the learning-by-doing of other stakeholders’ way of working is either non-existing or poor.

Lack of relevant knowledge and stakeholders having problem expressing their needs was identified in the study by Cao & Ramesh (2008) to be reasons why companies did not pre-define requirements in their development projects. Some interviewees at Scania addressed these reasons
with similar statements. Firstly, some interviewees mentioned that stakeholders had problems knowing what they really wanted since their knowledge about the new function was limited. Secondly, since functional requirements were emphasized and developers lacked methods or tools to manage NFRs, the NFRs that were considered were most of the time tacitly included. There are some templates used by the development at Scania that considers some NFRs, however these are not described thoroughly or recognized by the other stakeholders. The study conducted by Cao & Ramesh (2008) indicates that these problems often led to user requirements being neglected and emphasis was put on core functionality. While elicitation of NFRs are difficult as indicated in the study by Doerr et al. (2005), this urge the need of methods to handle this.

6.2 Lack of time

Cysneiros & Yu (2004) describes time pressure because of deliveries being demanded as one key factor that caused some requirements to be prioritized down.

The results of this study confirms this statement and extends this by investigating underlying reasons, related to an individual’s work, to why time pressure occurs. Mainly, many respondents emphasized that they needed to fully focus on their own deliverables and thereby did not have time to consider other stakeholders’ requirements. To cope with this, there exist many different processes to follow and documents that need to be filled in. However, these were perceived as a time consuming processes rather than an aid that would be worth the time spent on it in. Additionally, the respondents reveal an indication that this is a question of experience. Time pressure leads to focus on the individual’s most critical tasks, whereas this might lead to more time pressure in the future since neglecting other’s requirements may inflict complications later that would require time consuming changes. Realizing what impact the requirements from different stakeholder has on the deliverable is mentioned to be a vital part in working more proactive and thereby also avoiding late changes that are both costly and time consuming.

6.3 Requirement changes and administration

Volatile requirements are described in the theory as a problem area and the reasons for that are e.g. changes in the market environment (Davey & Cope, 2008), which are situations where non-functional requirements are commonly disregarded. In order to handle these types of situations, a dynamic approach, which is achievable by not perceiving a requirement as stable during the development phase, is mentioned to be preferred when the requirements can change frequently.

The respondents at Scania seems to be well aware of the volatility of their requirements, leading to that they do not see them as stable as they are claimed to be almost impossible to state clearly in the beginning of a project. Their method of handling this is to continue the development until the product is completed and verified to be compatible with other systems and first then specify the requirements that the product will achieve. A general standpoint for many respondents were that uncertain requirements will just be reworked in the future and thereby is only a waste of time to specify.

In other words, a dynamic approach to requirements is mentioned to assist in managing changes in the environment, whereas it can, at the same time, be a risk factor for elimination one of the most important aspects of requirement management, the documentation step, which is a crucial step when it comes to maintenance for example.
In this chapter, the results and analysis are discussed. Conclusions are drawn with emphasis on the research question presented in chapter 1.

7.1 Discussion

Herein are the results, analysis and the methodology discussed. The discussion has been conducted in sections as previous chapters.

7.1.1 Identifying and understanding other stakeholder needs

Stakeholders are, as Freeman (1984) described, important for the outcome of the project and covering their needs is essential. The first step in being able to follow a requirement process is thereby concluded to be a clear insight of what stakeholders that are involved in the development chain.

In this study, a vast majority of the respondents faced difficulties in expressing a generic overview of the involved stakeholders in the development process. One argument was that each project was unique to a certain degree, involving people from different departments at the organization, depending on what part of the vehicle that was being developed. However, the roles present in each project were described to be similar, disregarding of what department the involved persons came from, and the needs of the roles were somewhat hard to explicitly define for the respondents.

Forming a new way of working, while at the same time not putting enough emphasis on a critical step, the identification of the stakeholders involved, is indicated as a barrier for fully establishing this new process. This problem relates to the other problems highlighted as results in this study, where people seem to lack incentives to adapt to others and a more unstandardized, subjective way of working was developed. There seems to be a connection between how well one person can relate to another and how willing one is to put in extra effort to value the other’s needs.

As Zhao et al. (2011, p. 19) defined internal integration; “the degree to which a firm can structure its organizational practices, procedures and behaviors into collaborative, synchronized and manageable processes in order to fulfill customer requirements”, the term may arguably be used as a measurement of how efficient a requirement process can be within an organization. A process can be designed correctly and implemented throughout the organization, but how well it is managed over time is dependent on how the organization is understood by each individual. Integration should avoid different functions at the organization to be seen as independent.
Kauppinen et al. (2004) highlights people’s resistance to change as an implementation challenge, as people need a motivation factor. A similar challenge seems to exist after the implementation phase as well. In order to adapt a new way of working as a daily routine after trying it out, one seems to need to see the benefits continuously. An interesting indication by this study, in relation to the theory by Kauppinen et al. (2004) that a change has to be introduced at an individual level at first, then proceeded to a project level and lastly to an organizational level, is that to keep the change successful, the change has to be reflected back to an individual level after the implementation organizational-wide. The process should support and encourage employees to provide feedback that will enhance its continuous compatibility with the organizations overall processes.

An organization might benefit from creating a useful process for the employees, helping them to adapt to others affected by their work. However, even though all the necessary information about the process exists at hand for everyone to reach, problems still occur. While many of the guidelines today existed in documented form, the organization might benefit more from letting the employees experience critical problems in real-life situations. The respondents that had experience from working at different departments throughout the organization mentioned it as a more valuable experience than being told what needs that exists at a meeting or through a document. Therefore, a critical part, according to this study, of establishing a working process over time seems to be how it is distributed and educated among the employees, rather than the actual content of the process.

7.1.2 Lack of time

While user needs are, as suggested by Cysneiros & Yu (2004), usually prioritized down by time pressure, one intention of a structured method is to help the developer to be more efficient. It should assist the developer to guide the way throughout the development process in order to save time by doing the strategic choices for the developer.

It seems that people are driven by short term results, and adapting to a change will often result in a long term benefit, as illustrated by Kauppinen et al. (2004) as an implementation challenge. People feel that even though the need for a more structured way of working is needed, the adaption of a new process is yet a stressful moment for an individual.

In this case study, a requirement process has been implemented in the organization, yet the support the implemented process intends to provide is occasionally being perceived as time consuming. In other words, the intended purpose of the implemented process may have had a counter-effect for some employees.

The reasons behind this counter-effect may be because of several reasons. The most common factor, expressed by the respondents, for not working proactively was the lack of experience from earlier mistakes. People prioritize their own work efficiency and have a hard time seeing the work hours for the whole project and its contributions. Utilizing a proactive, new process might mean that one hour is spent more by the individual at the initial stage at the development chain, but at the same time this will prevent adjustments from being done at a later stage at the cost of three hours. However, employees lack incentives for making the necessary change, since they easier relate to their own work hours rather than the work effort of the whole project. In addition, a general feeling among the respondents was that time pressure will not subside if more time is spent on a current task. To solve this problem, one would have to make sure employees knew the consequences of neglecting some requirements initially but also provide support from a managerial level that emphasized the need to feel responsibility for the whole chain. This could
be concretized through workshops at other departments, common goals, review of earlier projects and more tools and resources for the initial requirement stages.

A paradox to the time pressure is that it seems to be partly caused by working reactively while at the same time be a barrier for adapting a way of working that is characterized by proactive actions. Working reactively means that errors in the design will be handled when they occur, rather than working proactively to prevent them from occurring. Late changes are those that are most costly and time consuming. As time pressure was stated to be frequently occurring, the employees strove to accomplish their goals as soon as possible, thinking this will do and moved on, essentially meaning that many aspects might not have been covered.

When the requirement process has been implemented, there is a clear indication from this study that there exist a need for a follow-up on what effect this method have had in the development process. Implementing it and expecting people to follow it is just not enough. As implied by Kauppinen et al. (2004), educating and involving developers in how the requirement process works and how and why it is important is vital. This would help to improve it but most of all, create a deeper understanding for how it works.

7.1.3 Requirements changes and administration

Due to the volatile nature of requirements, the documentation process of requirement in this study was often neglected until the function was implemented and ready. Similar to the study by Cao & Ramesh (2008), the requirements was not pre-defined but rather emerged as the projects progressed. It was clearly indicated that the requirements, from an external perspective and not from the function owner himself, was formulated very specifically and since it occurred that requirements specifications often lacked motivations, it can be assumed that it was mainly focus around pure functionality of the function.

Non-functional requirements, such as usability and maintainability, remains, as the literature study indicated (Glinz, 2007), a vital part of the outcome of a software development project. As the core functionality was in focus of many development projects in this case study, NFRs was disregarded. Trying to manage NFRs with structured methods while, at the same time, avoiding documenting them due to their complex nature is counter-productive. The reasons why developers avoid documenting NFRs seemed mainly to be because they are hard to measure and verify, not thought of and low prioritized. One way of dealing with these issues is, as Driessen & Hillebrand (2013) suggested, to address requirements as a capability, thus educating developers about these requirements which has been discussed earlier as a key solution. At the same time, having motivations for requirements and in particular NFRs is essential to manage changes at a late stage. If requirements are set without specific motivations and descriptions, others may not know how a change would affect other requirements or systems.

Moreover, it was identified that using the argument that the requirements will change as an excuse for not pre-defining the requirements and updating them frequently can pose problems in other areas. In a project group, there were representatives from many different departments at the organization. If some of them, e.g. groups in the end of the development chain, want to work proactively, they depend on input from the earlier stakeholders in the chain. If requirements have not been documented yet, then this will be troublesome. Simultaneously, the initial developer believes that these groups will just waste time by looking at unfinished, uncertain work. As requirements engineering is described to be an iterative process, one need to receive feedback frequently to keep the requirements updated and verified until they are set. A general
understanding is needed here, so that the project group works together towards a proactive approach.

7.1.4 Methodology

There are several aspects of the methodology that can be up for discussion. Firstly, the explorative studies that partly directed this study was at a limited departments and industry setting which could make it discussable to which degree this problem areas are applicable in other areas. Moreover, the interviews at Scania focused on the configuration process with parameters hence a very specific development area and therefore hard to generalize. It could also have been interesting to interview someone from a managerial position to see how they thought about this requirement process.

The stakeholder analysis was made after insights from both the explorative interviews and looking at their processes. Since they mentioned that every project was quite unique, one could have wanted more inputs from different departments and experience to cover as many project types as possible, thus extracting the an even better stakeholder map.

Having semi-structured interviews provided good insights and at the same time comparable data. In some cases, the respondents came unprepared even though the agenda was sent in advance.

The literature study which was continuously processed throughout this project was rather extensive and iterative. User needs and NFRs are well known areas of study, although few industry practices seem to appear. This was both an obstacle and a possibility, since we could contribute to this area by looking at these aspects that prevents developers to use structured methods for requirements engineering.

Furthermore, the vast amount of data from the analysis resulted in over one hundred pages of transcriptions. Categorizing the information and making use of it made us having to prioritize what to include and what to exclude. This could have led to useful information being neglected during the process.

Finally, the conclusions that were made are relevant but might be hard to implement. Some of them require effort from an organizational perspective in order to function properly. It would have been desirable to implement these changes and then review the effect, in order to conduct future studies.

Regarding the generalizability of this study, the modular environment at Scania contributed to that the process and product structure became rather complex in terms of managing requirements. The configuration of the developed software had to adapt to the modular development and its constraints and this, in turn, lead to that employees faced problems in satisfying different requirements that might interfere with each other. In a setting where the development is not driven by modularity, the configuration is not as dependent on design constrains, leading to that customizability might become easier and thereby easier to include various requirements from different stakeholders.
7.2 Conclusions

The conclusions that can be drawn from this study are presented in this chapter. These intend to answer the problem definition: What are the barriers preventing the establishment of structured methods within requirement management?

The identified barriers are presented below and these can be categorized under the four main phases of the framework of the requirement engineering process.

**The understanding of others is strongly associated to how you relate to them and their needs.** This barrier refers to the phase elicitation of requirements. An understanding of which stakeholders that are involved in the development chain appeared essential in order to extract user needs and to fully establish the requirement engineering process. This understanding should not remain the entry barrier in the implementation phase of a new process but rather be reflected back to continuously and encouraged by the project and organization, in order to keep the roles clearly established over time. The understanding of the organization was highly correlative with how much the individual expressed the value of satisfying other stakeholder needs. Employees with more knowledge about the organization and the processes were more positive to consider other stakeholder needs, an important starting point for establishing an efficient requirement engineering process.

**The establishment of a requirement engineering process is largely dependent on how it is distributed.** This barrier can be related to the documented phase. How well new processes for requirement engineering will be established within an organization is largely dependent on how it is distributed. Much emphasis was today put on the content of the actual, existing processes, whereas employees faced difficulties in both knowing about their existence as well as adapting to it if the distribution channels are inefficient. To avoid this, how knowledge is distributed and administrated needs to be evaluated and prioritized. This should be a continuous learning process since new tools and methods occur frequently.

**A reactive approach prevents others from being proactive.** Lastly, this barrier can be connected to the management phase. Requirements engineering remains an iterative process, essentially meaning that everyone involved needs to synchronize their actions according to each other in order for it to work. If some stakeholders lack the ability or the incentives to work proactively, then other stakeholders will be affected since they might depend on their information. In this scenario, the service market requested documents from development, even if not finished, in order to work proactively. However, the development group did not see the benefits of distributing work that was far from finished, as they thought it would mislead others. Here, a traditional waterfall mentality was established among the developers. They did not realize that it might be beneficial to show unfinished work in order to get early feedback that would prevent late changes, an important aspect of agile methods.
8 RECOMMENDATIONS AND FUTURE RESEARCH

In this chapter, recommendations on more detailed solutions and/or future work in this field are presented.

8.1 Recommendations

The recommendations are based on the discussion and conclusions that was drawn in the previous chapter. The recommendations will focus on how an organization in this specific context can improve its capability to use structured methods for requirement engineering.

**Develop methods and tools to identify stakeholders that are specific to your project.** Defining a clear role for every person that will be affected by the project is a vital part on the way to satisfy their different needs. As for today, employees have a hard time expressing a generic development chain and what different departments responsibilities are, leading to poor conditions for finding the incentives needed to adjust the daily work according to others. However, the departments at service market developed target customer descriptions in order to identify and describe their stakeholders and a similar approach could be beneficial for the development to implement.

**When new processes, tool and methods are launched, reflect on how they most efficiently can be distributed amongst all stakeholders.** How this is achieved should be reflected on, since documents regarding this are indicated to be problem area. The content inside the documents might be describing solutions to fulfill different requirements, yet the employees are not capable of knowing about their existence or extracting the information the documents contain. As for today, guideline documents may be created and distributed organizational-wide without follow-ups by the ones that implement it, leading to a subjective response by the receivers. Workshops, visits, observations etc. with different stakeholders has shown potential in this area according to some respondents, preferably with the ones that are always involved in your projects. When new directions are made in some sort of area then make a plan how this should be distributed.

**Make sure to educate and update the employees about the processes continuously.** How the current requirement management process is followed by each employee relies heavily on experience, while newly recruited face difficulties in both finding information regarding the existence of the processes and knowing which process to follow.
8.2 Future research

Future work refers to which areas that could be interesting to investigate more or complement this study with.

This thesis covered the main barriers that prevent employees to follow structured methods for requirement engineering. First of all, this study focused on the employee level and not on the organizational or managerial level. Making a study with focus on those aspects could be interesting areas to complement this study with, hence provide a broader set of perspectives concerning this matter. Economic consequences of these barriers could be more interesting from a managerial and organizational point of view.

Secondly, the barriers pointed out in this study were dependent on the incentives of the respondents. These incentives could be related to human factors and this opens up for psychological studies to complement this study, for example cognitive behaviors of employees.

Thirdly, the context was quite distinctive for this study, mostly since the configuration development was chosen. Performing a similar study with another context could reveal other barriers. The context of this study had a relatively complex and large development process. It would also be interesting to compare different contexts to see what they could benefit from each other.

Finally, solutions in how to prevent these barriers need to be further investigated and also practiced and evaluated.


APPENDIX A: EXPLORATIVE INTERVIEWS
GUIDELINE

Intervjuguide

Syftet med frågorna är att belysa vilka problem som existerar inom det aktuella området hos de olika avdelningarna för att lägga grund till den frågeställning som skall definieras. Olika frågor berör olika avdelningar på Scania.

Allmänna frågor

1) Hur bedrivs normalt förändringar i struktur eller arbetssätt i din organisation/avdelning?

2) Talar man om icke-funktionella krav på er avdelning? I vilket sammanhang? Finns det strukturerade metoder att hantera dessa?

3) När och hur kommer ni i kontakt med parametrar?

4) Har parametersättning eller hantering av icke-funktionella krav hanterats/diskuterats hos er som ett problemområde vid något tidigare tillfälle? Vad kom man fram till?

5) Upplevs parameterhantering som ett problem? När? T ex. Är antalet parametrar ett problem, ta hänsyn till olika aspekter när införandet av parametrar sker etc. (Test, verkstäder etc.)

6) Vilka slags problem kan ett stort antal parametrar bidra med? Kan komplexa system göras enklare? (Kombinatorik delas upp i enskilda parametrar?) ?) Hur verifierar man att alla täckts i testning?

7) Arbetas det aktivt med att göra funktioner kompatibla med framtida funktioner?
   a) Om ja: Talar man om utbyggnad m.a.p. data volym osv.?
   b) Om ja: Finns det riktlinjer?
   c) Om ja: Görs detta tillräckligt?
   d) Om nej: Vilka hinder finns?
   e) Om nej: Skapar detta extraarbete?

8) Vad innebär det för SOPS att produktstrukturen inte är definierad? (Olika beroenden, kommunikationsproblem, för mycket minne osv?) Finns det risk att projektets omfattning underskattas/överskattas i inledningen, dvs. lämnas det utrymme för utbyggnad/nedskalning i funktioner? Hur påverkar detta parametersättningen?

9) Avses bakåtkompabilitet?
   a) Om ja: Hur lätt är det att anpassa nya funktioner till gamla produkter?

10) Vad skulle, enligt er, vara drivkraften/skalet till att fokusera på att få till ett mer strukturerat arbetssätt med avseende på denna aspekten?
11) Hur anpassat är parametrarna för ändringar? Finns det riktlinjer för detta?

12) Hur sker dokumentering av ändringar? Är det ett problemområde och finns det behov av förbättringar?

13) Har man i åtanke hur ändringar i parametersättning påverkar helhetsstrukturen för funktionen?
   a) Hur undviker man att det påverkar andras arbete?

14) Inledning av projekt kan innebära vagt formulerade kravställningar, hur anpassas arbetet kring detta?

15) Kan krav beskrivas överflödigt? T ex. om det framgår på flera olika ställen kan det uppstå problem vid ändringar då det bara ändras på några av dessa.

16) Beskrivs krav/parametersättning självständigt? Kan beroenden upplevas som en svårighet vid behov av modifikation?

17) Går det enkelt att se hur systemet med parametrar fungerar och hänger ihop?

18) Hur lång tid tar det att förstå/sätta sig in i det?

19) Känns det som att det finns en gemensamt förståelse för parametersättning för alla berörda parter (utvecklare, kravsättare, test, eftermarknad etc.)

20) Finns det en tydlig struktur?

21) Är det enkelt att lokalisera och reparera problem med avseende på parametrar?

22) Är det enkelt att återställa fel till normalt läge?

23) Hur känsligt är det för förändringar?

24) Hur är kompabiliteten?

25) Hur ser det ut med garantier till kund?

26) Hur hanteras varianterna till parametrarna?

Avslutning

1) Vilka är de viktigaste aspekterna (NFR) att tänka på för er?
2) Har vi missat något?
Syfte

Syftet med den här intervjun är att se över hur kravhanteringen går till av icke-funktionella krav på Scania, främst mellan olika avdelningar som arbetar i olika steg i samma näringskedja. Vi ämnar titta på om man väger in olika avdelningars behov framgår i utvecklingsfasen, eller om det är mer funktionsorienterat där man snarare känner ett överlämningsansvar till nästkommande. Genom detta arbete hoppas vi kunna belysa vad som fungerar bra respektive dåligt med denna process för att på så sätt kunna ge rekommendationer för förbättring.

Kontext

Ämnet behandlar parametersättning och hur kraven på dessa hanteras. Fokus kommer främst att ligga utveckling av nya funktioner och implementeringen av dessa med avseende på parametersättning. Vi vill få djupare förståelse för hur parametrar sätts och vilka användare man tar hänsyn till när man sätter parametrar.

Inledning

1. Namn:
2. Yrkesroll:
3. Avdelning:
4. Hur bedrivs utvecklingsarbete på er avdelning?
   a. Finns det någon uttalad metod man följer? T ex. Agile, Waterfall, etc.
5. Vilka andra avdelningar samverkar ni med?
6. När och hur kommer ni i kontakt med parametrar? (ansvar, leveranser, program som används och hur det fungerar)
7. Kan ni ge exempel på när parametersättning har varit ett problem, alternativt lyckat?
8. Vilka behov har ni vad gäller parametersättning?
9. Vilka val gör man vid parametersättning och när sker detta?

Generellt om kravhantering med avseende på parametrar

1. Finns det en strukturerad metod för kravställningar/parametrar och dess identifiering av vilka behov som existerar?
   a. Inleds projekt med en kravställning/målsättning? Parametrar?
2. Sker kontinuerlig kontakt mellan kravställare och utvecklare för att säkerhetsställa att båda är med på att kraven tolkats lika?
   a. Framgår det vem kraven kommer ifrån och vilka behov som tagits i beaktande av kravställaren?
   b. Hur formuleras krav till andra och vilka är involverade?
   c. Hur kundorienterat anser du att tänket är på er avdelning, alternativt överlämningsfokuserat?
   d. Vilka dokument hamnar ni i kontakt med? UFR, AER etc.
   e. Är kravhanteringen ett problem och i såna fall vilka hinder finns det?

3. Hur arbetar ni mot andra avdelningar i utvecklingen? Avstämningar, mötesprotokoll, etc.
   a. Anser du att organisationsstrukturen är tydlig, dvs. hur din position påverkar andra avdelningar på Scania? Hur framgår det vem man ska vända sig till när något behöver korrigeras?
   b. Skiljer det sig hur man arbetar mot mål internt och mellan olika avdelningar? Uttalade metoder?
   c. Finns det något trade-off tänk i ert arbete?


5. Känner du att dina åsikter tas i beaktande i utvecklingsstadiet? Hur tas de emot? Varför?

**Dynamiska krav**

1. Hur hanteras krav när det sker ändringar på marknaden?
   a. Till vilken grad är kraven detaljerade och hur påverkar det ändringsbarheten?
   b. Hur arbetar man med verifiering av krav när de inte är fullt definierade?
   c. Verifieras det om krav fortfarande är aktuella?

2. Hur är det att gå tillbaka till ursprungliga krav vid ändringar, alternativt se motiveringar till varför ändringar har skett?
3. Sker det att krav sätts efter implementering?
   
a. Vilka konsekvenser kan detta ge? Har du något exempel?

4. Tänker ni på vilken kunskap ni delar med er av och när i utvecklingen det sker? Är det någon information du önskade att du haft tidigare eller saknat?

**Avslut**

1. Vad önskar du hade kunnat vara bättre vad gäller parametersättning generellt?

2. Något du vill tillägga som vi inte hunnit täcka?