

# **Problem Areas in Creating Valuable Smartphone Software**

A Software Product Management Case Study

Anna Kjölstad Svedu  
Johanna Leonard

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

Företagsekonomiska institutionen Ekonomihögskolan,  
Lunds Universitet  
Box 7080  
220 07 LUND

Institutionen för designvetenskaper,  
Lunds Tekniska Högskola  
LTH Box 118  
221 00 LUND

Copyright © Anna Kjölstad Svedu & Johanna Leonard

Examensarbeten i Technology Management - Nr 262/2014  
ISSN 1651-0100  
ISRN LUTVDG/TVTM--14/5262--/SE

E-husets tryckeri, Lund 2014  
Printed in Sweden

## Abstract

- Title:** Problem Areas in Creating Valuable Smartphone Software – A Software Product Management Case Study
- Authors:** Anna Kjölstad Svedu  
Johanna Leonard
- Supervisors:** Peter Svensson, Lund University School of Economics and Management  
Gudbjörg Erlingsdóttir, Faculty of Engineering, Lund University
- Issue of study:** Software product management is an immature research area, which takes a process perspective on value creation in the software context. The smartphone industry operates in a business-to-consumer setting. Software that is sold to an open market consisting of a mass of consumers is called market-driven software. Several authors have pointed out problem areas in either business-to-business companies or rather small business-to-consumer companies, not including embedded products. Furthermore, smartphones are categorized as embedded software products, meaning that it software is embedded in a non-software product, where software cannot be sold separately.
- Purpose:** The purpose of this thesis is to examine the problem areas a particular smartphone manufacturer encounters in its software product management, and thereby increase the understanding of software product management in the smartphone context.
- Methodology:** Based on the purpose of this study a qualitative case study was chosen. The research approach was open and explorative in its nature, in order to identify a wide range of problem areas. The data collection consisted mainly of interviews with employees at the case company.

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

**Conclusions:** The case company's process are described and analyzed. Several problem areas are identified in their way of working towards a more value-focused process. These problem areas are: *Lack of a common vision, Non-existing software product management role, Unpredictable and non-transparent requirement prioritization, Consumer awareness, but no consumer focus, Imbalance between idea generation and market analysis in elicitation activities* and *Operators are seen as the most important stakeholder*. They have to some extent been validated in the previous research.

The results can be used by researchers for identifying new research areas. There are also important guidelines for practitioners, and the results of the thesis can help them to understand the obstacles they face. Finally, the results enhance the understanding of previous industrial surveys on requirements engineering and software product management by validating results and adding a new perspective; the smartphone context.

**Keywords:** Smartphone, Software Product Management, Market-Driven Requirements Engineering, Release Planning, Case Study

## Glossary

**Embedded software** - A piece of software that is not sold as a stand-alone product, but is integrated in a non-software product.

**Feature** - An intentional distinguishing characteristic of a software item in performance or functionality.

**Requirement** - A singular documented physical and functional need that a particular design, product or process must be able to perform.

**Smartphone** - A powerful computing device offering traditional wireless voice service as well as native software applications and the ability to connect to and run a myriad of Internet-based services including e-mail, geo-location, streaming video, and social networking, while providing a good user experience.

**Software evolution** - The process of updating a software repeatedly as a means of increasing functionality.

**Software product** - A product whose primary component is software.

**Software release** - A specific version of a software.

**Stakeholder** - A person, a group of people or an organization with an interest or concern in something.

**User interface** - The space that a person interacts with on their smartphone.

**Value** - *From a consumer point of view:* the difference between the consumer's perceived benefits and the costs.

*From a business point of view:* the profit generated by product sales.

## Acknowledgement

First of all, we would like to thank our case company for giving us the opportunity to come and study your work. We would very much like to thank you for welcoming us with open arms and being very open and honest about your issues, we appreciate that this must not always have been easy. We would like to especially thank our company hosts. We are grateful for your support and input.

Secondly, we would like to thank our University supervisors, Gudbjörg Erlingsdóttir and Peter Svensson, for your patience and your support. These past few months have not always been easy, and it has been very valuable for us to be able to come ask your advice.

Writing this thesis has been an amazing opportunity to learn about areas where we, quite honestly, were absolute beginners. It has been as challenging as it has been rewarding.

Lund, May 2014  
Anna Kjölstad Svedu  
Johanna Leonard

## Table of Content

<b>1</b>	<b>INTRODUCTION .....</b>	<b>10</b>
1.1	BACKGROUND .....	10
1.1.1	<i>Software Products .....</i>	<i>10</i>
1.1.1	<i>Market-driven Software .....</i>	<i>11</i>
1.1.2	<i>Software Product Management .....</i>	<i>12</i>
1.1.3	<i>The Smartphone Industry .....</i>	<i>13</i>
1.2	ISSUE OF STUDY .....	14
1.3	PURPOSE.....	15
1.3.1	<i>Research Questions .....</i>	<i>15</i>
1.4	LIMITATIONS.....	15
1.5	OUTLINE OF THE REPORT.....	16
<b>2</b>	<b>METHODOLOGY.....</b>	<b>18</b>
2.1	INTRODUCTION.....	18
2.2	RESEARCH APPROACH.....	18
2.2.1	<i>A Single Case Study.....</i>	<i>18</i>
2.2.2	<i>Open and Explorative Approach To Capture a Full Picture .....</i>	<i>19</i>
2.3	RESEARCH PROCESS.....	20
2.3.1	<i>Phase I – Pre-study .....</i>	<i>21</i>
2.3.2	<i>Phase II – Qualitative Data Collection and Theoretical Framework .....</i>	<i>21</i>
2.3.3	<i>Phase III – Analysis.....</i>	<i>25</i>
2.3.4	<i>Phase IV – Discussion &amp; Conclusions .....</i>	<i>26</i>
2.4	CONSEQUENCES OF THE RESEARCH METHOD.....	26
2.4.1	<i>Validity.....</i>	<i>26</i>
2.4.2	<i>Reliability .....</i>	<i>27</i>
2.4.3	<i>Generalizability.....</i>	<i>28</i>
<b>3</b>	<b>THEORETICAL FRAMEWORK.....</b>	<b>29</b>
3.1	INTRODUCTION.....	29
1.1.2	<i>Definitions.....</i>	<i>29</i>
3.2	SOFTWARE PRODUCT MANAGEMENT .....	30
3.2.1	<i>Defining Software Product Management .....</i>	<i>31</i>
3.2.2	<i>The Role of the Software Product Manager .....</i>	<i>32</i>
3.2.3	<i>The Contents of Software Product Management .....</i>	<i>33</i>
3.2.4	<i>Stakeholders in Software Product Management .....</i>	<i>36</i>
3.2.5	<i>The Effects of Insufficient Software Product Management .....</i>	<i>39</i>
3.3	REQUIREMENTS ENGINEERING .....	42
3.3.1	<i>The Characteristics of Market-Driven Requirements Engineering .....</i>	<i>44</i>
3.3.2	<i>Activities in the Requirements Engineering Process .....</i>	<i>46</i>
3.3.3	<i>Common Challenges in Requirements Engineering .....</i>	<i>48</i>
3.4	SUMMARY OF THEORETICAL FRAMEWORK .....	50
<b>4</b>	<b>CASE STUDY FINDINGS.....</b>	<b>51</b>

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

4.1	INTRODUCTION.....	51
4.2	THE COMPANY CONTEXT.....	51
4.3	THE INTERVIEWEES.....	52
4.4	OVERVIEW OF SOFTWARE DEVELOPMENT PROCESSES AT THE COMPANY.....	53
4.4.1	<i>The Software Development Process</i> .....	53
4.4.2	<i>New Functionality Process</i> .....	54
4.4.3	<i>The Quality Improvement Process</i> .....	57
4.5	PROBLEM AREAS AT THE CASE COMPANY.....	60
4.5.1	<i>Lack of a Common Vision</i> .....	61
4.5.2	<i>Non-existing Software Product Management Role</i> .....	64
4.5.3	<i>Unpredictable and Non-transparent Requirement Prioritization</i> .....	69
4.5.4	<i>Consumer Awareness, But No Consumer Focus</i> .....	74
4.5.5	<i>Imbalance Between Idea Generation and Market Analysis in Elicitation Activities</i> 79	
4.5.6	<i>Operators Are Seen as the Most Important Stakeholder</i> .....	83
4.6	SUMMARY OF THE CASE STUDY FINDINGS.....	86
<b>5</b>	<b>CONCLUSIONS AND FINAL REMARKS</b> .....	<b>89</b>
5.1	INTRODUCTION.....	89
5.2	RECOMMENDATIONS FOR THE COMPANY.....	90
5.3	DISCUSSION OF RESULTS.....	91
5.3.1	<i>Theoretical Contribution</i> .....	92
5.3.2	<i>Suggestions for Future Research</i> .....	93
5.4	CONCLUDING REMARK.....	94
<b>6</b>	<b>BIBLIOGRAPHY</b> .....	<b>95</b>
<b>7</b>	<b>APPENDIX I - INTERVIEW GUIDE</b> .....	<b>99</b>



## List of Figures

Figure 1. Number of articles with "software product management" in the title, abstract, keywords, or refer to such publications (Fricker, 2012).....	12
Figure 2. The outline of the report.....	17
Figure 3. The research process used in the thesis.....	20
Figure 4. The general construct of the theoretical framework.....	24
Figure 5. The theoretical framework. ....	29
Figure 6. The lifecycle spanning nature of software product management (Ebert & Brinkkemper, 2014, p. 2).....	34
Figure 7. A reference framework for software product management (van de Weerd et al., 2006, p. 4).....	35
Figure 8. Main activities in Van de Weerd et al. (2006) framework regarding Software Product Management.....	35
Figure 9. The results of insufficient software product management (Ebert, 2009, p. 16). ....	40
Figure 10. Summary of root causes and tangible problems caused by insufficient software product management adapted from Ebert and Brinkkemper (2014); Bennet and Rajlich (2000); Ebert (2007); Ebert (2009) .....	41
Figure 11. The theoretical framework. ....	42
Figure 12. The three main activities in requirements engineering. ....	46
Figure 13. The software development process at the Company. ....	53
Figure 14. The new functionality process. ....	55
Figure 15. The quality improvement process.....	58
Figure 16. The software product manager aspects that we have found and not found within the Company (Fricker, 2012; Bekkers et al., 2010; Ebert, 2007; Van De Weerd et al., 2006). ....	68
Figure 17. Problem areas connected to the three main activities in software product management.....	88

## List of Tables

Table 1. The ISPMA software product management framework (ISPMA, 2013, p. 9). ....	34
Table 2. Problems in requirements engineering. ....	49
Table 3. Summary of the persons interviewed and their organization.....	53
Table 4. Table showing main stakeholders in the theoretical framework, compared to the involved stakeholders at the Company, their main activities, the name of the organization representing the stakeholder in the Company and how much impact they have.....	61
Table 5. Summary of problem areas, related to previous research.....	87
Table 6. Summarizing table over possible remedies to the Company, divided upon the found problem areas, and on long-term and short-term. ....	91

# 1 Introduction

---

*This chapter introduces the reader to the subject of this master thesis. The first sections describe the background to the problem. Firstly, by introducing the case company and how the research focus was selected. Secondly, by introducing the reader to software products and the related research areas. Thirdly, we discuss the current conditions in the smartphone industry. After this the purpose is presented, along with research questions and limitations to the study. Finally, the outline of the report is presented.*

---

## 1.1 Background

This master thesis was conducted as a case study at a smartphone manufacturer. The case company, from hereon referred to as the Company, was previously a large player in the smartphone industry, but has in recent years experienced a significant decline in market share. When we first arrived at the Company, they suggested that we examined how software updates should be designed to improve consumer satisfaction and loyalty in terms of increased retention rates, since they had experienced severe problems with getting consumers to repurchase their products. However, after having conducted a pre-study, we decided to shift our focus, mainly for two reasons. Firstly, we assessed that the market changed rapidly, and consumer preferences along with it. Therefore, we did not believe that results from first proposal would be valid over a longer period of time. Secondly, during the pre-study we began to understand that the process was loosely defined, and that the interviewees pointed out several internal problem areas. Essentially, we believed that we could deliver more valuable results by studying the process for defining software releases. Below, we discuss the product and industry aspects that led us to our choice of purpose.

### 1.1.1 Software Products

Software has become an important part of our daily lives and is now embedded in almost all technical products. Software is not only found in smartphones, but also in

airplanes, cars and computers (Karlsson, Regnell, Natt och Dag, Dahlstedt, & Persson, 2007).

Kittlaus and Clough (2009) state that software is a manifestation of human know-how, and is therefore in essence the result of the knowledge of the people behind it. Furthermore, software is easily distributed and duplicated, which is both an advantage and a disadvantage (Kittlaus & Clough, 2009, p. 5). Kittlaus and Clough (2009) define a *software product* as:

*A product whose primary component is software.*  
(Kittlaus and Clough, 2009, p. 6)

Additionally, the same authors define *embedded software* as:

*A piece of software that is not sold as a stand-alone product, but is integrated in a non-software product.*  
(Kittlaus and Clough, 2009, p. 6)

Embedded software is only one part of the product, which is closely integrated with hardware components (Kittlaus and Clough, 2009, p. 6). Consequently, product success is driven by the over-all functionality of the integrated sub-components, rather than the software itself (Kittlaus and Clough, 2009, p. 6).

### **1.1.1 Market-driven Software**

Software that is sold to an open market consisting of a mass of consumers is commonly referred to as market-driven software. In a market-driven software development process there are both general and specific challenges to address. Several studies aim at highlighting the specific problems connected to such processes. For example, Karlsson et al. (2007) summarize the key problem areas requirements engineering and point out problem areas that are specific to market-driven organizations. Regnell and Brinkkemper (2005), similarly compile a list of seven specific challenges in market-driven requirements engineering. Based on these two studies, some key problem areas are: the balance between technology push and market pull, marketing and development not effectively communicating and a constant flow of loosely defined requirements that are supposed to represent millions of individual needs and desires (Karlsson et al., 2007; Regnell and Brinkkemper, 2005).

## 1.1.2 Software Product Management

The increased presence of software products in our daily lives is reflected in the increased academic interest in the software research area. Subsequently, the research area of *software product management* has grown in popularity over the past decade, as can be seen in Figure 1 below.

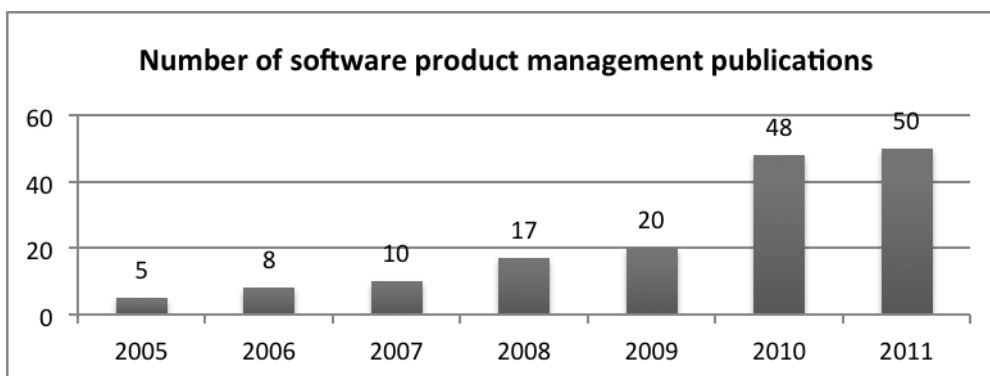


Figure 1. Number of articles with "software product management" in the title, abstract, keywords, or refer to such publications (Fricker, 2012).

Software product management is described as the processing of issues involving requirements, products and releases in an organized manner (Van De Weerd, Brinkkemper, Nieuwenhuis, Versendaal, & Bijlsma, 2006). The over-all rationale of software product management is to generate profit by achieving continuous success over the software product's life cycle (Kittlaus & Clough, 2009). Moreover, Barney, Aurum, and Wohlin (2008) claim the there is insufficient alignment between the product, project and business perspectives in the software context, which accentuates the need for the research discipline.

In general, software companies operate in highly competitive environments, where time-to-market is a primary product development criterion. In this type of competitive environment, where customers are well informed, Anderson and Narus (1998) argue the importance of *knowing what customers value*. However, Boehm (2006) has shown that software engineering traditionally does not focus on value-creation.

It is apparent that software is more important today than ever, but also that it is a complex product to manage. Additionally, traditional software engineering has not focused on value-creation to a very large extent, but in the current competitive climate the life cycle needs to be managed in order to be successful.

### 1.1.3 The Smartphone Industry

In order to understand the fundamental nature of the industry, it is important to understand that the conditions have changed since the breakthrough of the smartphone. Let us first define what a smartphone is. Kenney and Pon (2011) define smartphones as:

*...powerful computing devices offering traditional wireless voice service as well as native software applications and , perhaps most importantly, the ability to connect to and run a myriad of Internet-based services including e-mail, geo-location, streaming video, and social networking, while providing a good user experience.*

(Kenney & Pon, 2011, p. 240)

As can be interpreted from this definition, a smartphone is a complex product, where several aspects need to work together in order to provide a good user experience. These include everything from software, to hardware and third-party applications.

In order to gain market shares, smartphone makers need to improve how they manage consumer relationships and their brand, since consumers choose from a variety of firms that offer similar features (Dedrick, Kraemer, & Linden, 2011). The consumers' preferences have changed from preferring operators' own designed services to favoring a market-driven ecosystem of applications (Hacklin, Battistini, & Von Krogh, 2013). Consumers can choose exactly which applications and functionalities they have in their smartphones, thus creating their own user experience to a large extent (Hacklin et al., 2013).

However, the introduction of the smartphone did not just challenge manufacturers, it also revealed new opportunities. Smartphone manufacturers now have the possibility to increase consumers' perception of the product throughout the lifetime of the smartphone through software updates. Black (2006) suggests that, in order to become and remain competitive, smartphone manufacturers need to keep pace

with consumer expectations and the changing environment by updating smartphone software. Furthermore, many bloggers and tech journalists argue the importance of updating smartphones to the latest software, partly for security reasons, and partly because of the appreciation of new features and user interface updates (Grabham, 2012; Smith-Strickland, 2013). Hartmann, Trew and Bosch (2012) claim that consumers choose their next smartphone not depending on hardware, but mostly on the software it contains.

It is apparent that the telecom industry has changed over past few years. Moreover, this shift has resulted in an increased software focus. Therefore, it is reasonable that manufacturers want to seize the opportunity to produce superior products by using software updates.

## **1.2 Issue of Study**

The Company faces obstacles in creating valuable products that may be related to the specific attributes of the product and the market-driven context. Firstly, smartphones are categorized as embedded software products, meaning that its software is embedded in a non-software product, where software cannot be sold separately (Kittlaus & Clough, 2009). In smartphones, software is closely integrated with hardware, and the functionality of the entire system will drive the requirements process. This makes matters complex from a product management perspective. Secondly, the Company operates in a market-driven context (see Karlsson et al., 2007 and Regnell and Brinkkemper, 2005), which poses problems in managing the flow of requirements.

We claim that it is important to examine how the Company could produce more valuable products. It is reasonable to assume that this could be achieved by having a better knowledge about market requirements, and a more value-focused process, which connects this thesis to both market-driven requirements engineering and software product management.

Software product management takes a process perspective on value creation in the software context. However, it is an immature research area as Maglyas, Nikula, and Smolander (2011) stated in their mapping of the state of knowledge in software product management:

*...software product management has not been studied fully and this area offers many opportunities for research. The importance of conducting studies in software product management cannot be overestimated because SPM offers tools and practices for achieving business goals as well as increasing the predictability and profitability of developing software products and services.*

*(Maglyas, Nikula, & Smolander, 2013, p. 34)*

According to Karlsson et al. (2007) there is a need to identify problem areas at large- and medium-sized companies producing embedded products. C. Ebert and Brinkkemper (2014) base their results on a large study of business-to-business companies. Nevertheless, software product management in a business-to-consumer setting has not yet been as extensively evaluated. Therefore, we state that it is an interesting addition to software product management research to study a large smartphone manufacturer from this perspective.

## **1.3 Purpose**

The purpose of this thesis is to examine the problem areas a particular smartphone manufacturer encounters in its software product management, and thereby increase the understanding of software product management in the smartphone context.

### **1.3.1 Research Questions**

In order to answer the purpose of this study, we have developed the following research questions:

- How do the processes at the Company relate to the processes described in Software Product Management?
- Which are the problem areas that the Company encounters from a Software Product Management and Market-Driven Requirements Engineering perspective?

## **1.4 Limitations**

- The study has been conducted at a software development site with a European focus
- We have not studied the Company in relation to its parent company

- The only product, which has been studied, is the smartphone. We have disregarded other products and accessories sold by the company
- We have studied exclusively the software development processes, and have thereby not studied the hardware development processes
- We have not studied the operational development, that is, we have not interviewed programmers on their work. The aim of this study has been to take the management perspective and examine the decisions that are made and the guidelines that are set up. Subsequently, we have only interviewed people at a middle management level
- We have focused on the three last phases of software product management: product roadmapping, requirements engineering and release planning excluding product portfolio management
- In release planning, we do not discuss specification and validation, as this is not conducted at the management level where we have focused the study

## 1.5 Outline of the Report

The remainder of this thesis is organized as follows. In Chapter 2 the research methodology is explained. Related work and the theoretical framework are presented in Chapter 3. The case study findings are presented and compared to the theoretical framework in Chapter 4. Chapter 5 presents the conclusions from the study, and discusses the contribution to the company as well as academia. The outline of the report is summarized in Figure 2 below.



Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

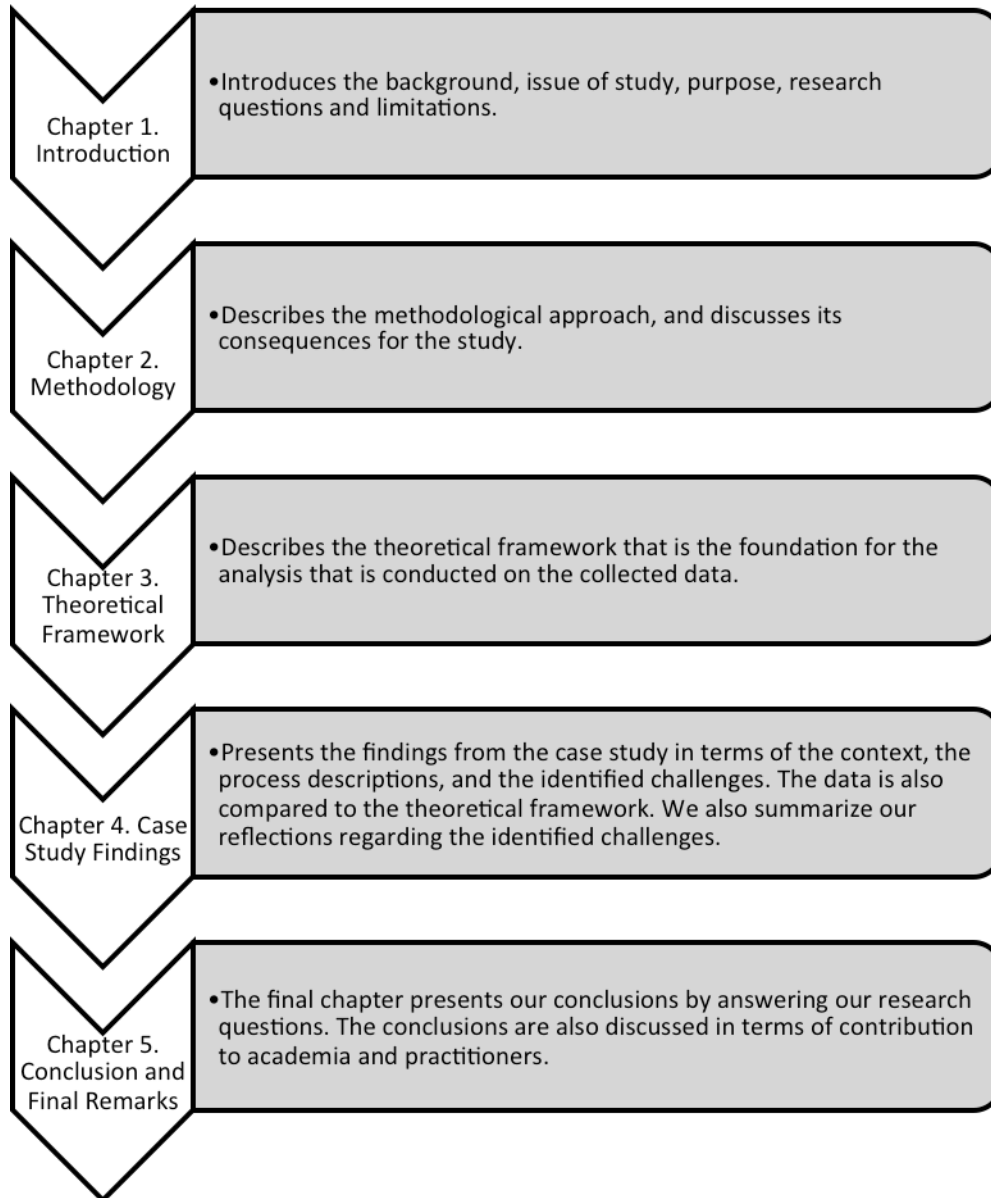


Figure 2. The outline of the report.

## 2 Methodology

---

*This chapter describes the methodology used by the authors to conduct the study. First, the general approach is described in terms of research approach. Thereafter, the research process is described, and this section also addresses data collection and the construction of the theoretical framework. Lastly, the consequences for validity, reliability and generalizability are discussed.*

---

### 2.1 Introduction

The purpose of this chapter is to describe and motivate the choices that we made regarding the methodology of this thesis. We chose to conduct a single case study with qualitative data collection through interviews and observations. Furthermore, we chose an open approach where we collected data and constructed the theoretical framework in parallel. The starting point of this thesis is the case company, and the data collected there is used to validate some findings in previous research, and this perspective is represented in our choice of methodology.

### 2.2 Research Approach

#### 2.2.1 A Single Case Study

The purpose of this study is to study how a certain company works with certain processes. Therefore, the choice to conduct a single case study was the natural choice for us. According to Yin (2007), a case study approach is suitable when studying a phenomenon in its usual context, when it is difficult to determine the difference between the phenomenon and context, and moreover, when the researcher does not control the research environment. This was true for our research purpose. The Company is the starting point of our study, and as such a case study is the obvious choice, especially since we aim at exploring their particular problem areas and obstacles.

Furthermore, we aimed at increasing the understanding of software product management in a particular context. Case studies are especially appropriate for such purposes according to Jacobsen (2002).

One option would have been to benchmark different case companies against each other, this would have provided us with more general results. However, due to time and accessibility constraints, this was not possible. Therefore, we chose to conduct a single case study, where we explored the nature of the processes at a case company, in order to later on benchmark against literature that we found relevant. We did this with the purpose of gaining in-depth understanding of the Company's way of working with the processes that we examined. Consequently, we have not been able to generalize the result to a very large extent (Lundahl & Skärvad, 1999). The generalizability of the study is further discussed in section 2.3.3.

### **2.2.2 Open and Explorative Approach To Capture a Full Picture**

When conducting our study, we have used an open approach, since we wanted to ensure that we received a comprehensive description of the situation at the company. We felt that closing our approach would have limited our results too much. Also, we were initially not very knowledgeable in the processes at the company, and therefore it proved difficult to create a theoretical framework that we felt was sufficiently comprehensive. Therefore, we chose an open approach, where the case company is the starting point, and theoretical perspectives were added as we found them in the data collection.

The base of our findings is the case company, and the problem areas that we found there were mapped against problem areas identified in previous research. This enabled us to form our own conclusions, not limiting ourselves solely to problems that are identified in literature, but also opened up the possibility of finding new perspectives. Moreover, it ensured that we did not seek to confirm previous expectations, but enabled us to keep a relatively open mind. In this setting, we found this to be necessary in order to add value for the company as well as academia.

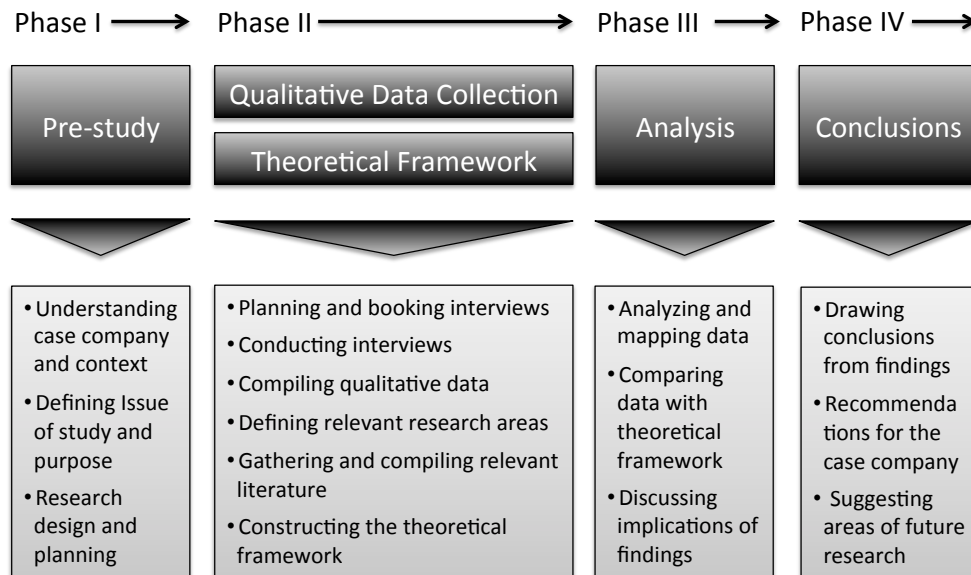
When we arrived at the Company, we were not completely sure what we would find. Therefore our approach became explorative. This was well in line with the open approach described previously in this section. Also, since we were not very knowledgeable in the research areas or the processes at the company, it was a necessity for us to extensively explore both previous research and the ways of the

Company. Otherwise, it would have been very difficult for us to find meaning in our results.

Software product management as a research area is still quite immature (Fricker, 2012; Maglyas et al., 2011) and when researching an immature research area an explorative study is suitable according to Yin (2003). These aspects validated our choice of approach.

## 2.3 Research Process

This section describes the general research process used for conducting the study. It covers the entire process from the initial pre-study until the finalized report. Due to the explorative and open approach that we chose, some phases overlapped, especially data collection and theoretical framework. Below, the phase activities and output is presented. Figure 3, below, is a summary of the research process used in the thesis. For each phase, the main activities are listed in the corresponding box.



**Figure 3. The research process used in the thesis.**

### **2.3.1 Phase I – Pre-study**

Initially, the problem that we were to study was not clearly defined. Therefore, we decided to conduct a pre-study in order to define our purpose and research questions. In situations without clearly predefined problems, Lundahl & Skärvad (1999) recommend the use of pre-studies.

The aim with the pre-study was to gain a deeper understanding of the case-company business and the setting in which it operates. Three interviews were conducted in order to uncover the process of software updates, the stakeholders involved and the general business conditions in the industry. The interviews were semi-structured and asked the interviewees to explain the process for creating software updates. We selected three roles representing three different organizations involved in the processes.

As we soon understood, the processes themselves were loosely defined, even to the people involved in them. Furthermore, the three people that we interviewed had already pointed out several problem areas in the processes, that they regarded as important to address. As a result we concluded that exploring the processes and benchmarking them against previous literature would constitute a significant contribution to the case company. During the pre-study, we also began initial literature studies. Here we discovered that the research areas connected to the issues unveiled in the pre-study were quite immature, and we concluded that a case study in the company context would also constitute an academic contribution.

The main output of the pre-study was a clearer understanding of the company context and an indication of where their problems lay. This enabled us to define the issue of study and purpose.

### **2.3.2 Phase II – Qualitative Data Collection and Theoretical Framework**

The main activities of the second phase was collecting data and constructing the theoretical framework. These activities would in turn form the basis for the analysis. They were done in parallel as a result of the open and explorative research approach described earlier. The following sections describe the methods for collecting data and constructing the theoretical framework.

The output of the second phase was the compiled qualitative data and the theoretical framework. That is, the material we needed for analyzing the results in the following phase.

### **Qualitative Data Collection**

We chose to collect qualitative data for the purposes of this study. It was the method that was clearly most suitable due to the open and explorative approach. It would have been impossible for us to collect quantitative data since we would not know what to look for. Furthermore, we aimed at gaining in-depth knowledge of a social phenomenon in a certain context, and according to Jacobsen (2002), qualitative data is most suitable in such cases. Karlsson et al. (2007) assert that qualitative methods are seldom used in software engineering research, but that qualitative approaches fit well in this context.

Moreover, we were interested in the interviewees' perception of the processes, and there could be no other way of capturing this except by using qualitative data. One reason why we wanted to capture their perception of the processes was that we suspected that the different stakeholders had very different perspectives and subsequently different perceptions. The consequence is that results are not as easily replicated, which is discussed later.

### **Interviews**

The qualitative interviews were rather open, though not completely, as they were conducted with a theme and a fixed order of questions (Jacobsen, 2002). The purpose of using an open approach was to not limit the respondents, while still ensuring that the interview focused around the specific theme (Jacobsen, 2002). All the interviews were conducted in person, with both of us present. The interviews lasted between 30 minutes and one hour, depending on how talkative the respondent was.

#### *Selection*

The interviewees were selected based on their immediate participation in the development process. At the end of each interview, the interviewees were asked whether they had suggestions for other roles to include in the data collection, in order to obtain as comprehensive data as possible. However, roles not involved in the current process are not included; this was a strict limitation in the data

collection, enforced in order to ensure relevant input describing the current process as is, rather than how it should be. We decided to only interview persons at middle management level, as we wanted to examine their decisions and the guidelines that they created. As a result, we were not able to fully uncover whether these guidelines are actually followed. Also, there might have been other problems that we were not able to identify. Another consequence is that we did not get a complete view of the process, but only the management perspective, which was also reasonable as software product management concerns management issues.

#### *Interview Guide*

The interview guide was constructed by drawing inspiration from guides used in previous studies. In particular, we used the interview guide used by Barney et al. (2008), as we found this to fit what we wanted to achieve. Moreover, we included very open questions where we asked the interviewee to describe how they viewed the processes. The interview guide is found in Appendix I. It should be noted that it was not always strictly followed; we deliberately chose to let the interviewees be free to follow their own trains of thought and associations, since we aimed at understanding their perspective. However, we endeavored to make sure that all the relevant questions were answered. When we found that we had more questions or needed to clarify something, we asked supplementary questions via e-mail or in person.

#### *Documentation*

Both of us were present during all the interviews. One person was responsible for asking questions, and the other one was responsible for taking notes during the interview. All the interviews were recorded, and afterwards they were listened to and the notes were completed and citations were transcribed. All the interviews were conducted in Swedish, and the results included in the report have therefore been translated.

#### **Observations**

We have spent approximately four months at the case company, and during this time we have been seated in the open office space. This has enabled us to understand the business environment. We have also had the opportunity to observe some meetings, which has further contributed to our understanding of how they work with updates.

## Theoretical Framework

The purpose of the theoretical framework was to provide a theoretical perspective of our findings, to the extent that this was possible. Thus, it was used as a benchmark of sorts. This enabled us to later analyze our results by comparing findings to previous literature. The theoretical framework was mostly based on two research areas: software product management and requirements engineering. We based the theoretical framework on the general processes in these areas, as well as previously identified problem areas.

We chose software product management as the base for the theoretical framework since the level of detail fit the processes we studied at the Company well. When discussing software, we mean both the application software and operating system. Since we saw an issue that the Company's products did not succeed in the market, we wanted to examine the software product management, as the purpose of the area is to generate more profitable products.

Requirements engineering is a set of sub activities in software product management. It was used as a complement to software product management in the theoretical framework. The data collection revealed significant problems in this particular area of software product management. Thus, this area is analyzed more in-depth in this study, whereas other areas such as portfolio planning and road mapping are only touched upon. We have focused on the area of market-driven requirements engineering because of the business-to-consumer context at the Company.

The general construct of the theoretical framework is summarized in Figure 4 below.

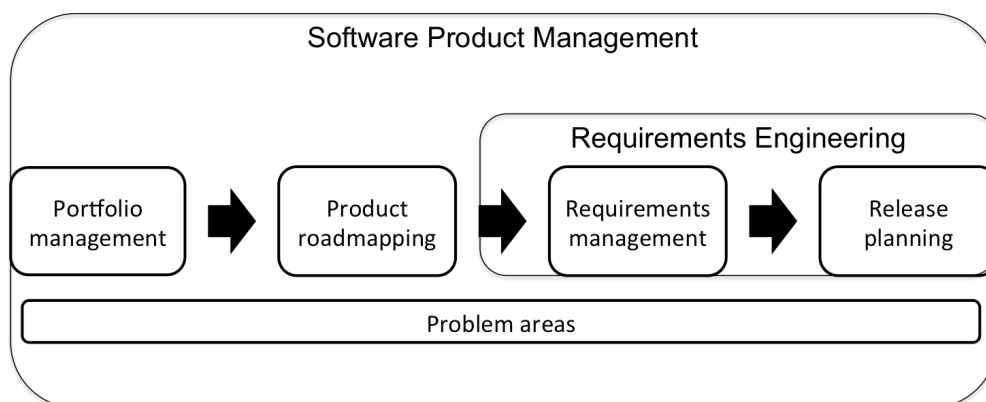


Figure 4. The general construct of the theoretical framework.



### ***Constructing the Theoretical Framework***

In order to ensure that the theoretical framework was based on reliable sources, we used almost exclusively scientific articles and books. Consequently, we have been able to form a reliable framework. We used the Internet to find articles and books, by searching primarily the database *LUBsearch*<sup>1</sup>, but also *Google Scholar*. Most of the articles utilized in the framework were peer-reviewed scientific publications, which means that the reliability has been ensured to some extent.

The immaturity of the research area posed some problems. Firstly, there were a limited number of scholars that had been published within the subject, meaning that the material itself was limited. Nevertheless, we managed to review most of the literature and are confident that we included sufficient perspectives. Moreover, we found that the majority of the research was conducted in a business-to-business setting, and therefore it needed to be completed with market-driven requirements engineering in order for us to know if it was applicable in the business-to-consumer context. It is our view that despite the bulk of previous research being conducted in business-to-business contexts, it was applicable in the business-to-consumer context when completed with market-driven requirements engineering.

### **2.3.3 Phase III – Analysis**

The main activity of the fourth phase was analyzing the data collected in Phase III. Initially, the data was compared and categorized, in order to ease the analysis process. This was conducted by reading transcripts and categorizing different sections from them after certain headlines and themes, a methodology suggested by Jacobsen (2002). When the results were systematized, we began interpreting the data and grouping different categories, also according to the process suggested by Jacobsen (2002). Then, the data was compared to the theoretical framework, in order to find similarities and differences.

The output of Phase III was a mapping of how we interpreted the process that the interviewees had described to us, and a compilation of different problem areas in the processes, as well as a comparison with the existing theory.

---

<sup>1</sup> Lund University library search engine.

### **2.3.4 Phase IV – Discussion & Conclusions**

Based on the analysis performed in Phase III, the results were discussed in terms of possible reasons for similarities and differences with literature, as well as what could be the possible consequences and causes for the discovered issues. Moreover, the issues were discussed in terms of what was special to the studied context. Also, based on the analysis and discussion the research conclusions were compiled and documented. The research conclusions were divided in the contribution for the case company, the contribution to academia, suggestions for further research, as well as the possibilities of generalizing the results.

As a part of the final phase of the study, efforts were also made to ensure a proper handover of the practical implications and recommendations for the case company. This effort consisted of workshops and presentations at the case company. The output of the final phase was the finalized report.

## **2.4 Consequences of the Research Method**

The selected research method had consequences for the results of the study. In this section, we discuss the consequences of the selected methodology in terms of how valid and reliable the results are.

### **2.4.1 Validity**

Yin (2013) describes a valid study as one that has collected and interpreted data in such a way that the conclusions adequately describe the studied organization.

In order to ensure validity, we adopted an intensive in-depth approach, where we spent the good part of four months at the case company. This enabled us to gain an in-depth understanding of the setting. However, it might be a source of bias in our process. We managed this by using theoretical perspectives to ensure objectivity. We claim that it was important for us to be close to our research subjects, as this enabled us to understand their day-to-day work.

We interviewed several people with different roles, which contributed to a nuanced picture of the process. We aimed at describing the interviewee's perception of the process by not steering their answers too much, and we believe that in doing so we

ourselves perceived the process as they actually experienced it. Moreover, we completed the interviews with observing meetings and discussions, which contributed to the internal validity of the study. Furthermore, we were not content with the interviewees' accounts of how they use their reporting systems, but insisted on seeing them for ourselves.

As for the methodology of this study, we turned to previously conducted research within the area and the methodology used in those studies. As a result, we confirmed that the methodology was applicable in the research area. Moreover, the interview guide used was composed after quite extensive research in the area, and was to a large extent inspired by an interview guide previously used and tested by experienced researchers. Moreover, we tested our interview guide on a practitioner and revised it accordingly, before we started conducting the interviews.

When analyzing the results, the findings were compared to problem areas found in previous research. Therefore we assess that the findings are to some extent grounded in previous theory.

### **2.4.2 Reliability**

Reliability concerns whether the study has been conducted without apparent errors or bias, that is, if it has been conducted in a way, which would enable it to be conducted again with the same results (Jacobsen, 2002).

We were both present during the interviews, in order to ensure that we perceived and understood the answers correctly. We claim that this reduced the risk of biased results. Moreover, one person was responsible for taking notes during the interviews, ensuring that we had documented material of all interviews. Also, we recorded all the interviews in order to be able to go back and listen to the interviews in retrospect. We processed all the material by reading the notes and listening to the recording, in order to ensure good documentation. Both of us categorized the material after it was compiled, ensuring at least two pairs of eyes. Furthermore, when things were unclear, we requested clarifications and on occasion, we conducted an additional interview. We also consulted our hosts at the Company on our findings. Nevertheless, since we used a qualitative approach, data was interpreted on several levels. This affects the reliability of the study.

During the process, we used our academic supervisors as sounding boards, in order to check the findings and to improve our process. Moreover, we took notes in a joint document every day, which enabled us to follow up on the progress of our work, as well as easing knowledge transfer.

### **2.4.3 Generalizability**

Generalizing qualitative research is a difficult matter, as it is typically focused on in-depth studies aimed at describing the context and significance of what has been studied, and it has been stated that the transferability is in such cases an empirical issue (Bryman & Bell, 2007).

The aim of this study was not to generate generalizable results, but rather to richly describe a phenomenon in a very particular context. Therefore, we do not claim that the results are generalizable in the sense that they can be attributed to other contexts than the case setting. However, the results might apply in settings that are in some sense similar and we believe that the descriptions in this report are sufficiently rich for a reader to assess whether they might be applicable. Nevertheless, this is a matter of further research.

## 3 Theoretical framework

---

*This chapter gives the reader an understanding of the central theories in the research area, which together constitute the theoretical framework. This is later used to clarify and compare the results found in the interviews, and is further discussed in the analysis.*

---

### 3.1 Introduction

The theoretical framework is described in figure 5. The theoretical framework is a guide throughout the thesis and facilitates the understanding of our findings. It is also to label some of the processes and problem areas.

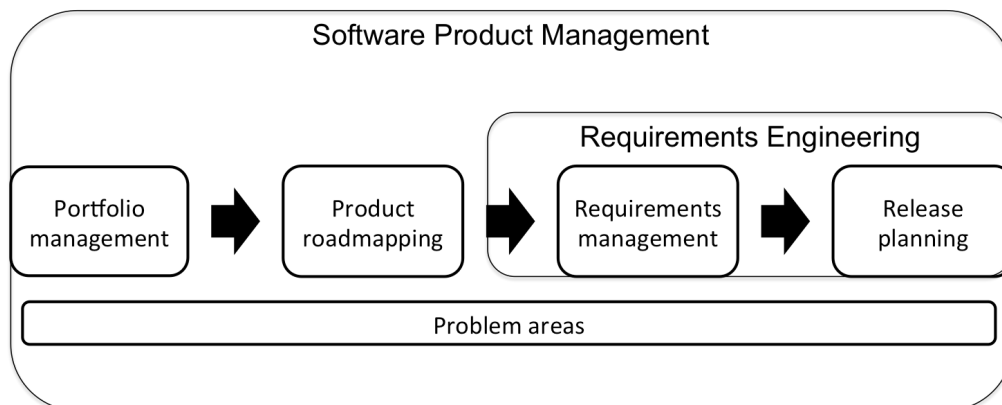


Figure 5. The theoretical framework.

#### 1.1.2 Definitions

The most important words used in this thesis are defined in order to increase the understanding for the reader.

We define stakeholders as a person, group or organization that has interest or concern in an organization.

Barney et al. (2008) define product value in a software product context, as being comprised by three aspects, product value, customer's perceived value and relationship value. The value of a product is directly linked to the competitors' value

and enhances if the product has higher quality than the competitor and decreases if the product has lower quality than the competitor (Alwis, Hlupic, & Fitzgerald, 2003). The consumer's perceived value is usually explained as the perceived benefits divided by perceived price, and is directly linked to competing products (Weinstein, 1999). The social relationship between the consumer and the company creates the relationship value, and is accessed through the two former values (Henneberg, Pardo, Mouzas, & Naudé, 2005) .

Barney et al. (2008) argue that three value perspectives are important for developers to have in mind when developing software, and these should also be linked with decisions made in the business, product and project level:

- **Business perspective:** *Business value stems from product sale.*
- **Value for product perspective:** *Product value stems from Customer and Market requirements.*
- **Value for project perspective:** *Project value stems from project budget/timing/delivery etc.*

(Barney et al., 2008, p. 580)

## 3.2 Software Product Management

Many traditional (non-software) companies have implemented a product manager and noticed several positive effects, such as improved product success and healthier internal competitiveness (Murphy & Gorchels, 1996). Nevertheless, software product management is a relatively young and immature research area, but it has received increased attention over the past decade (Fricker, 2012; Maglyas et al., 2011). The *International Software Product Management Association* (ISPMA) was founded in 2009, in order to foster software product management. This is one demonstration of the increased attention being paid to the research area, in both academia and in the industry (ISPMA, 2013).

Moreover, it has been observed that there are several positive effects of implementing software product management; significantly improving duration (time to market), schedule adherence and hand-over quality (Ebert, 2007). Fricker (2012) claims that a special product management area is required for software as it differs from conventional products. Software is relatively easy to change, can be duplicated

at low cost and will be released in several stages. Therefore, managers need to not only focus on the next release but also have a long-term evolutionary perspective (Wohlin & Aurum, 2005).

### 3.2.1 Defining Software Product Management

Since the research area is young and immature, we have tried to be consistent in the use of the same definitions through out the thesis. There is no general definition of software product management, thus we chose one that fits the purpose of the thesis.

Software product management is described as the organized process of handling issues involving requirements, products and releases in the product development process (Van De Weerd et al., 2006). The overall rationale of software product management is to generate profit, by achieving continuous success over the software product's life cycle (Kittlaus & Clough, 2009).

In their mapping of the research area of software product management, Maglyas et al. (2011) conclude that there is no universally accepted definition of software product management, as the area is still quite immature research-wise. However, software product management is frequently been defined as:

*"the discipline and role, which governs a product (or solution or service) from its inception to the market/customer delivery in order to generate biggest possible value to the business"*

(Ebert, 2007; Fricker, 2012; Maglyas et al, 2011)

Fricker (2012) highlights that this definition covers three crucial aims: the governance of software, the full software life cycle perspective and the generation of business value. Kittlaus and Clough (2009) similarly define software product management as *"the management of a software product (or product family or platform) over its entire life cycle in accordance with corporate level objectives"*.

As there is no universally accepted single definition of software product management, we chose Ebert's (2007) definition:

*The discipline and role, which governs a product (or solution or service) from its inception to the market/customer delivery in order to generate biggest possible value to the business.*

*(Christof Ebert, 2007, p. 851)*

This definition was chosen based on the frequent citations found in subsequent literature and our opinion that it captures the essential software product management aspects of governance, value creation, and the full life cycle perspective frequently highlighted in research.

### **3.2.2 The Role of the Software Product Manager**

The software product manager is the “mini-CEO” of a particular software product (Bekkers, van de Weerd, Spruit, & Brinkkemper, 2010; Fricker, 2012; Maglyas et al., 2013), acting as a representative for the product both internally and externally (Fricker, 2012). He or she represents what the product does, what it is, whom it serves, and what it means to the company and customers (Fricker, 2012). Essentially, the software product manager drives and enables successful evolution of the company portfolio (Fricker, 2012) and he or she evaluates which functions to retain, evolve and terminate (Ebert, 2007). This, however, is a challenging task as it involves compromising between different needs of the market, customers and other stakeholders, and aligning the entities around the value creation (Ebert, 2007; Bekkers et al., 2010).

The product manager works in three dimensions: creating and winning product and business case, conquering markets and increasing market share, and delivering value to customers (Ebert, 2007). He or she develops and aligns the product strategy with the company strategy and market needs, and through coordination between departments such as product development, marketing, sales, distribution, service, and support, he or she implements the strategy plans (Fricker, 2012; Ebert, 2007). In effect, this means serving as an interface between business functions and technical functions. Also, the software product manager ensures that the product strategy aims at creating and retaining economic success (Ebert, 2007).

Moreover, Van De Weerd et al. (2006) define the software product manager responsibilities as managing requirements, defining releases and defining products in a context comprising of several internal and external stakeholder involvements.



Ebert (2007) makes a similar statement, stating that the product manager governs the life cycle of a product. Therefore he or she is ultimately responsible for requirements, release definition, product release lifecycles, forming cross-functional introduction teams and preparing and implementing the business case, and these responsibilities exceed the single project. According to Fricker (2012), the software product manager also participates in innovation and takes leadership of new product ideas.

### 3.2.3 The Contents of Software Product Management

Several frameworks have been developed by authors to explain the activities in software product management. This section exhibits the most frequently used, and explains the activities it contains to give an overview of software product management and the activities included in the area.

Ebert & Brinkkemper (2014) emphasize that software product management spans the entire product lifecycle, from strategy and concept, through market entry and development, to product evolution. Their lifecycle perspective is illustrated in figure 6 below. Software product management should also align the three value perspectives, product, project and business.

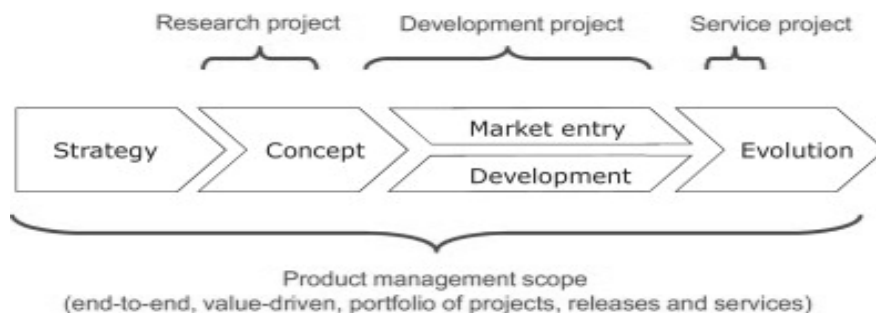


Figure 6. The lifecycle spanning nature of software product management (Ebert & Brinkkemper, 2014, p. 2).

The most important research results to this date are summarized in the ISPMA Software Product Management Framework (table 1), which is based on previous framework of Ebert (2007), Kittlaus & Clough (2009) and Van De Weerd et al. (2006). The grey areas are the software product manager's direct responsibility. According to Maglyas et al. (2013) this framework represents integration between previous

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

frameworks, and can be seen as a consensus of researchers and practitioners within the area of software product management. The three frameworks the ISPMA framework is based upon highlight and emphasize different aspects of software product management.

Strategic Management	Product Strategy	Product Planning	Development	Marketing	Sales and Distribution	Service and Support
Corporate Strategy	Positioning and Product Definition	Product Life-Cycle Management	Engineering Management	Marketing Planning	Sales Planning	Service Planning and Preparation
Portfolio Management	Delivery model and Service Strategy	Roadmapping	Project Management	Customer Analysis	Channel Preparation	Service Provisioning
Innovation Management	Sourcing	Release Planning	Project Requirements Engineering	Opportunity Management	Customer Relationship Management	Technical Support
Resource Management	Business Case and Costing	Product Requirements Engineering	User Experience Design	Marketing Mix Optimization	Operational Sales	Marketing Support
Market Analysis	Pricing		Quality Management	Product Launches	Operational Distribution	Sales Support
Product Analysis	Ecosystem Management			Operational Marketing		
	Legal and IPR Management					
	Performance and Risk Management					
<b>Participation</b>	<b>Core SPM</b>		<b>Orchestration</b>			

**Table 1. The ISPMA software product management framework (ISPMA, 2013, p. 9).**

The framework of van de Weerd et al. (2006) emphasizes the input to and output from these processes, especially with regards to the stakeholders involved.

## Problem Areas in Creating Valuable Smartphone Software - A Software Product Management Case Study

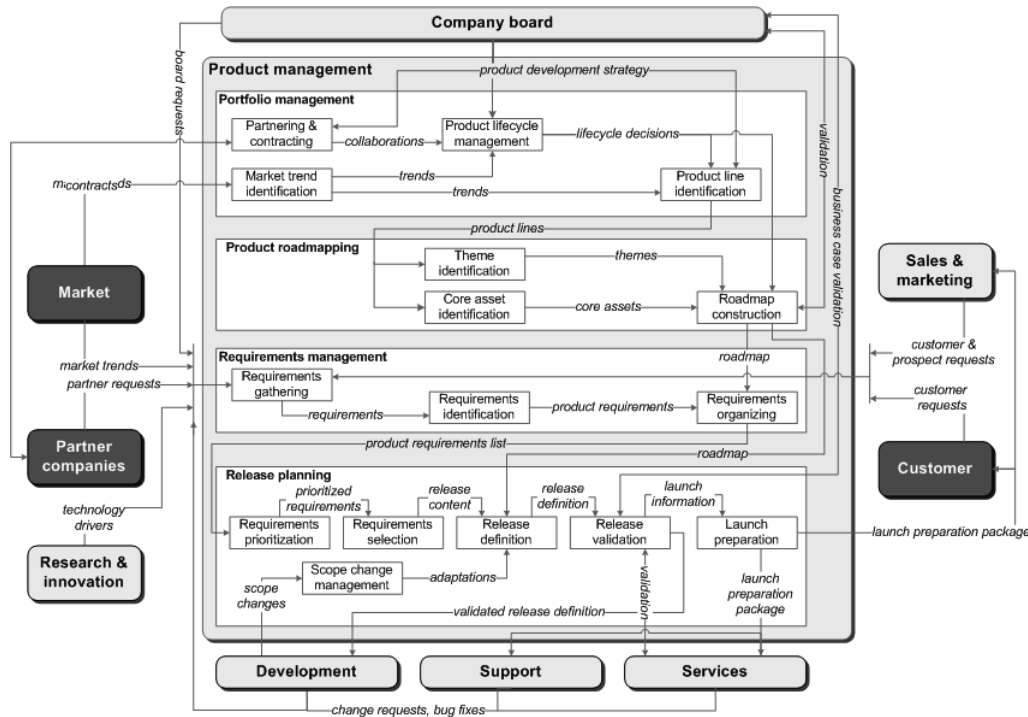


Figure 7. A reference framework for software product management (van de Weerd et al., 2006, p. 4)

For the purposes of this thesis, we chose Van De Weerd et al.'s (2006) framework as a reference framework for analyzing findings, since it clearly highlights the stakeholder perspective and as the level of detail suits the perspective of the thesis very well.

Van De Weerd et al. (2006) define four main sub-processes of software product management: portfolio management, product roadmapping, requirements management and release planning, see figure 8.

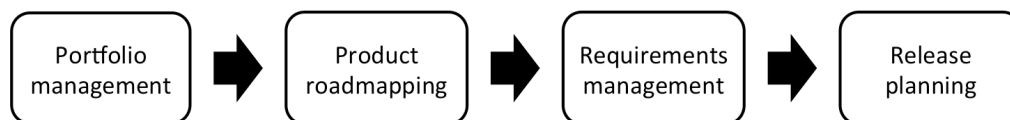


Figure 8. Main activities in Van de Weerd et al. (2006) framework regarding Software Product Management

Portfolio management concerns decision making about the portfolio of products, identifying new business opportunities, deciding about the product lifecycle and establishing partnerships and contracts (Van De Weerd et al., 2006).

Product roadmapping, on the other hand, concerns identifying and communicating the use of scientific and technological resources. Van De Weerd et al. (2006) determine that portfolio management gives input to the roadmapping process, where themes and core assets are identified, which in turn could be used later on in the requirements processes.

Requirements management is the activities consisting of gathering, identifying, revising and organizing the incoming requirements from different stakeholders (Van de Weerd et al., 2006).

Release planning is concerned with making the software available to the users. First, requirements are prioritized, and the product requirements that are to be implemented in the next release are selected (Van de Weerd et al., 2006). Since requirements management and release planning is requirements engineering, it will be described in section 3.3.

### **3.2.4 Stakeholders in Software Product Management**

The stakeholders in software product management have an active role in many of the stages in the process. This section summarizes the key stakeholders in software product management, and further aims at explaining their relation to, and impact on, software product management. Stakeholders are divided into two categories: external and internal stakeholders, where internal stakeholders are other functions within the company and external stakeholders are stakeholders found outside of the company.

Since the product manager has great responsibility in regards to product functionality, but has no formal authority over the development team, decision-making in software product management is a complex process, which requires the commitment of many stakeholders (van de Weerd et al, 2006). Moreover, the software product manager needs to coordinate internal stakeholders in order to align them around the company processes and activities. Successful coordination results in the product, as well as stakeholder activities are aligned with the company strategy, the market situation and stakeholder interests (Fricker, 2012). According to Ebert (2007), one reason for failing software product management is the lack of commitment from the internal stakeholders. Bekkers et al. (2010) frequently

highlight the formalized involvement of both internal and external stakeholders as a sign of mature software product management practices.

### ***Internal Stakeholders***

*Higher level management* (represented by the company board) is responsible for defining and communicating the overall strategy, vision and mission (van de Weerd et al, 2006), as well as product development strategies and resource allocation (Fricker, 2012). These stakeholders are main sources of input when constructing portfolio management strategies, which are then translated into product roadmaps. Furthermore, higher level management can be a source of requests during requirements management (Fricker, 2012). The product manager develops product strategies in agreement with top management (Maglyas et al, 2013).

The stakeholder *Research & Development* (or Research & Innovation as defined by van de Weerd (2006)) identifies new opportunities for product and technology innovation (Fricker, 2012; van de Weerd et al, 2006). The organization defines how new opportunities are to be integrated into the products (Fricker, 2012). They provide input into the requirements management activity, namely in the sub activity of requirements gathering (van de Weerd et al, 2006).

*Development* is a major stakeholder, working as a receiver of input from other stakeholders. According to Fricker (2012), development is mainly represented by the project manager, who is responsible for executing a project, which eventually leads to a product release. This involves managing the project scope, in terms of feasibility and implementation, and handing over the result to the product manager. In collaboration with development, the product manager plans product scope and monitors product evolution (Fricker, 2012).

*Sales & Marketing* is the generally the primary interface to the customer (van de Weerd et al, 2006; Kittlaus & Clough, 2009) and should represent the voice of the customer, by providing access to customers (Fricker, 2012). Thus, being a major source for new requests (van de Weerd et al, 2006). Moreover, this stakeholder is crucial in order to ensure that the product addresses the needs of the market (Fricker, 2012). As a representative for the market, sales and marketing could also provide insights on market trends as input to portfolio management (van de Weerd

et al, 2006). The product manager plans the marketing mix and monitors product success with the marketing department (Fricker, 2012).

*Service* (van de Weerd et al, 2006), or *distribution* (Fricker, 2012), is responsible for making the product available for consumption. In a business-to-business context, this entails implementing the software at the customer organization. This stakeholder provides input to requirements engineering in the form of change requests and bug fixes, and is in release planning responsible for launch preparation.

*Support* is involved in providing helpdesk services (van de Weerd et al, 2006), user training and consulting (Fricker, 2012), as well as minor maintenance in the terms of finding and reporting defects (van de Weerd et al, 2006; Fricker, 2012). They provide input to requirements engineering by submitting change requests and bug fixes (van de Weerd et al, 2006). Kittlaus & Clough (2009) claim that support could be a very helpful source of process input, especially for consumer products, since they represent a major interface to current customers, and receive first-hand information as to how the products are being received in the market.

Fricker (2012) also defines *finance* and *legal* as internal stakeholders, albeit with a more passive role than the others discussed in this paper. König (2009) especially points out finance as an important key stakeholder. Finance needs to ensure that sufficient revenue streams are realized and is also concerned with the allocation of resources and costs (Fricker, 2012). Legal are concerned with license use and protecting products from infringement and competitors (Fricker, 2012).

Ebert (2007) defines the core team of software product management, including the product manager, marketing and project manager (development), as the three most influential and important internal stakeholders.

### **External Stakeholders**

Van de Weerd et al (2006) defines potential customers and competitors as the stakeholder *The Market*. Fricker (2012) uses a similar notation and includes competitors, providers of substitute products, and new entrants, but not potential customers. Input from the market is instrumental in portfolio management and requirements management as a means to integrate market trends in the process (van de Weerd et al, 2006; Kittlaus & Clough, 2009).

*Partners* are an important external stakeholder as they resell and distribute software products, provide referrals and even cooperate in development (van de Weerd et al, 2006; Fricker, 2012). Partners provide input to portfolio management in terms of contracts, and to requirements management in terms of partner requests. Kittlaus & Clough (2009) emphasize that partners are even more important in the case of consumer products when the internal sales and distribution team have limited or no direct end-customer contact. Fricker (2012) further includes suppliers of components, platforms and systems as a stakeholder on whom the company depends, as its effort builds on the features and quality of the input provided by suppliers.

Current *customers* are sources of feature requests, which is usually communicated and included in requirements management by internal stakeholders such as support, sales and marketing and service (van de Weerd et al, 2006). Kittlaus & Clough (2009) suggest that user groups and customers could be a very valuable source of input in product-related areas, if they are representative for the desired market segments. A common reason for failing software projects is the misunderstanding and misinterpretation of customer needs and adapting accordingly (Ebert, 2007).

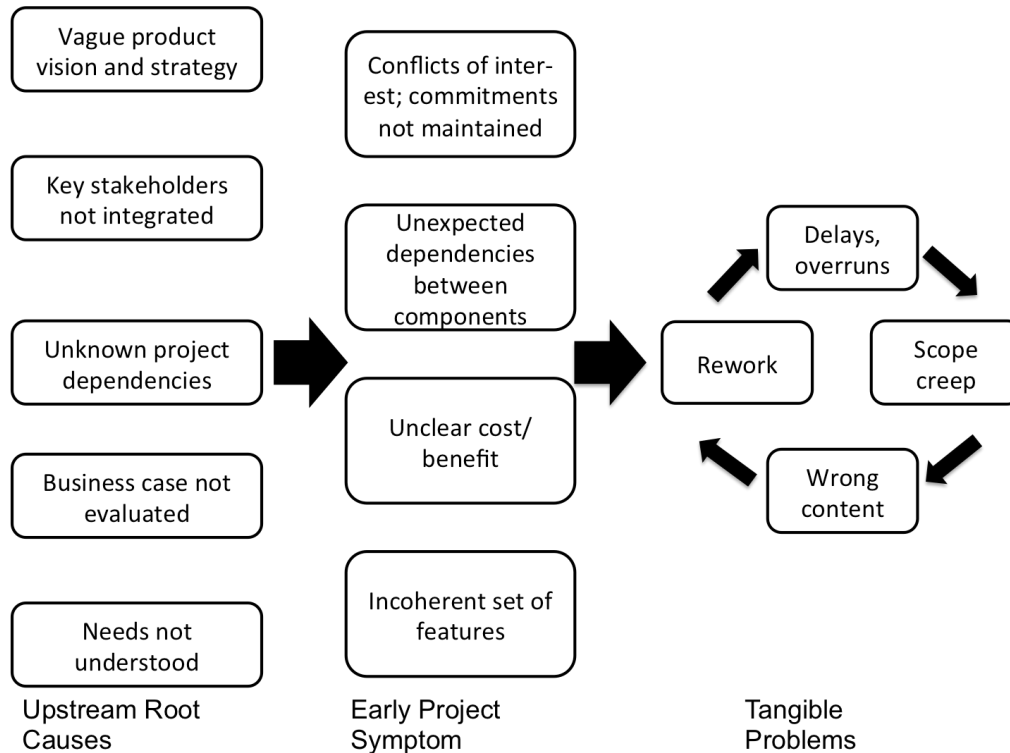
Many stakeholders are active in the process and play a vital role, without them there would be no input, or execution for that matter. The stakeholders' roles and involvement will later on be used as a foundation when analyzing the stakeholder involvement at the Company.

### **3.2.5 The Effects of Insufficient Software Product Management**

Ebert (2007) concludes that it has been frequently observed that: "companies win or fail depending on their product managers". Ebert (2009) claims that the typical problems caused by product management insufficiencies are lack of vision, vague market and business understanding, and the disintegration of key stakeholders. Such problems result in conflicts of interest regarding priorities and content, as well as incomplete requirements. This inevitably leads to rework, causing delays, which subsequently causes scope creep, and a vicious circle is initiated. This situation is described in Figure 9 below.

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---



**Figure 9. The results of insufficient software product management (Ebert, 2009, p. 16).**

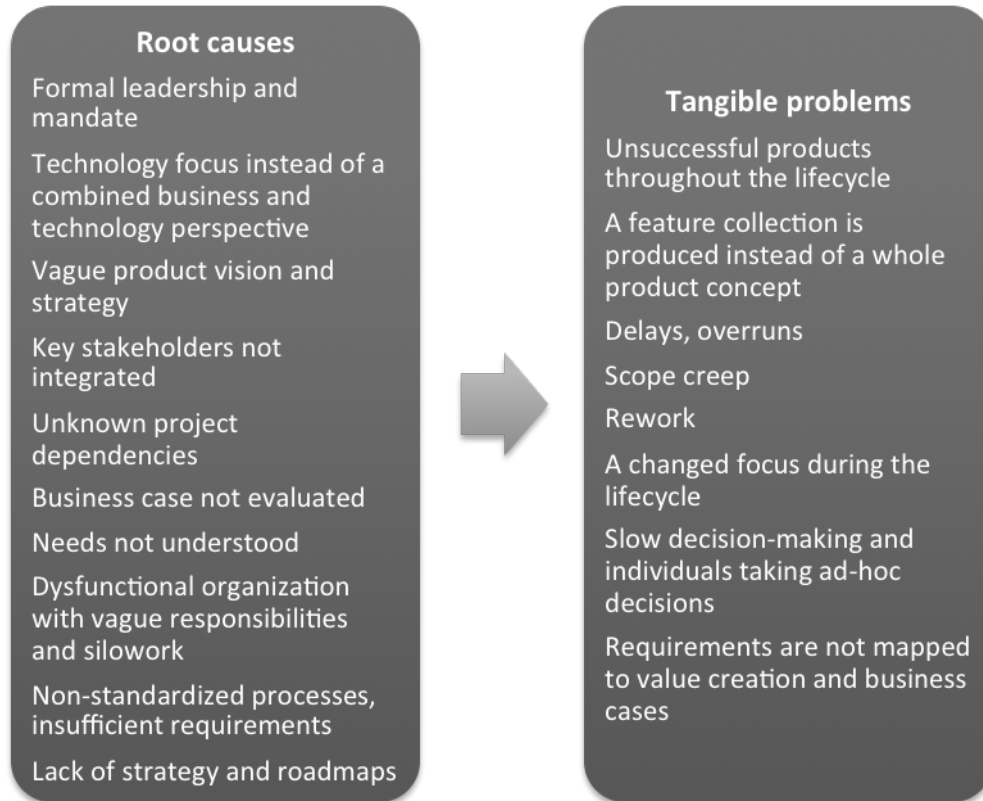
Furthermore, Ebert and Brinkkemper (2014) point out that too much focus on technology instead of a combined business and technology perspective results in a feature collection instead of a whole product concept, this is also seen by Rajlich and Bennett (2000) who emphasize the need of a business perspective. Also, the lack of a formal leadership and mandate is seen as significant obstacles for achieving product success throughout the lifecycle by Ebert (2007).

Ebert and Brinkkemper (2014) list four additional root causes to problem areas in software product management; *dysfunctional organization with vague responsibilities and silowork, non-standardized processes, insufficient requirements and lack of strategy and roadmaps*. The resulting problems of these root causes are: changing focus during the lifecycle, slow decision-making, individuals taking ad-hoc decisions and requirements not being mapped to value creation and business cases.



Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---



**Figure 10. Summary of root causes and tangible problems caused by insufficient software product management adapted from Ebert and Brinkkemper (2014); Bennet and Rajlich (2000); Ebert (2007); Ebert (2009)**

This section is summarized in figure 10, it shows that many severe problems arises when the software product management is incomplete. The aim is to further use the causes and problems to see how the Company is working according to software product management, and what problems they face.

### 3.3 Requirements Engineering

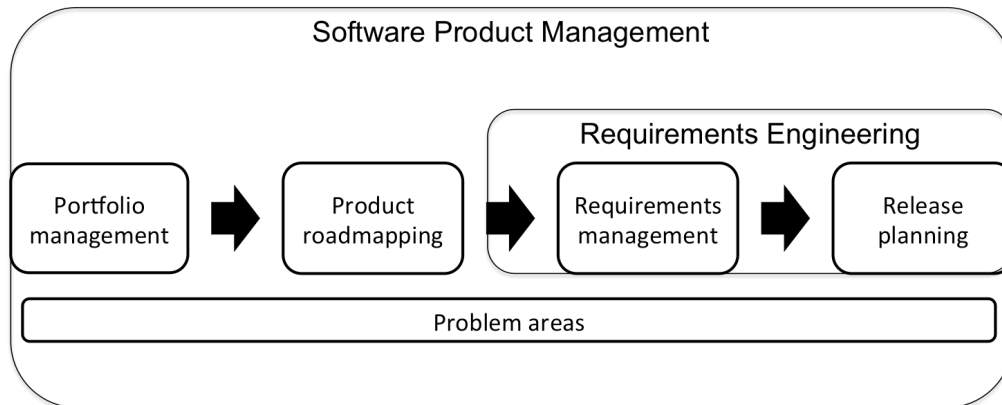


Figure 11. The theoretical framework.

In this section we further examine some of the requirements engineering activities as shown in figure 11. The requirements engineering process is significant for assuring quality in a software product process (Grunbacher & Briggs, 2001). The activities performed in requirements engineering are: elicitation of requirements from stakeholders, specification of requirements, validation of requirements, prioritization of requirements and release planning. It can be compared to requirements management and release planning in van de Weerd et al.'s (2006) framework. These activities are done in order to understand what the stakeholders want from the software, thereafter the requirements are prioritized in order to be able to develop valuable software. A requirement specifies the implementation in a system, and describes the behavior of the system (Lawrence, Wiegers, & Ebert, 2001).

Several authors claim that requirements engineering can be used as a tool to improve the value-creation in a software development process (Barney et al., 2008; C. Ebert & Brinkemper, 2014; Kittlaus & Clough, 2009). However, developers are used to work in a value neutral way and it can be hard for them to adjust (Aurum & Wohlin, 2007). The aim of a value creating requirements engineering process is to align requirements from consumers with the strategic business objectives (Aurum & Wohlin, 2007).

The requirements engineering activities include the stakeholders identified in van de Weerd et al., (2006) framework in several ways. The stakeholders define value from

## Problem Areas in Creating Valuable Smartphone Software - A Software Product Management Case Study

---

their different points of view (Keller, 1995). Thus, information and knowledge transfer among these should be included to have an effective process (Wohlin & Aurum, 2005). Moreover, Aurum and Wohlin (2007) emphasize the importance to involve several stakeholders in the process to include various perspectives of product, project and business value.

Haron and Sahibuddin (2010) claim that in order to prevent high maintenance costs, requirements need to be understood early in the process. This mindset is shared by Favaro (2002) who argue the importance of having a robust requirement process to decrease costs later in the development process.

It is important to outperform competitors in the requirements engineering process, and improve processes continuously since the quality of the process determines the product's level of quality (Regnell & Brinkkemper, 2005). Additionally, when designing a requirements engineering process for a company it is important to consider the company's situational factors, such as the company's, product's and customer's level of maturity (Regnell & Brinkkemper, 2005).

In conclusion, it is important to have a robust requirements engineering process to secure the process output. Furthermore, the most important things to keep in mind in the requirements engineering activities are:

- To align the market needs, the product definition and the strategic business goals
- Create value through the involvement of all relevant stakeholder point of view
- To have a better requirements engineering process than competitors'
- To take into account important requirements early in the process to avoid high maintenance costs

This is done in order to create a valuable and competitive product, and decrease the software's maintenance costs.

### **3.3.1 The Characteristics of Market-Driven Requirements Engineering**

Since the Company operates within a market-driven context it is important to understand the implications in their requirements engineering. Thus, the main differences between market-driven and customer-specific requirements engineering are described in this section.

#### ***Communication with customers is difficult***

When offering software to an open market instead of a specific customer, software development is referred to as market-driven (Regnell & Brinkkemper, 2005). In customer-specific development one develops a special solution to one customer, considering only this one customer's requirements. In a market-driven context there are several, sometimes millions of consumers with different requirements (Regnell & Brinkkemper, 2005). One of the specific differences is managing the constant flow of new requirements from the big group of customers (Karlsson et al., 2007; Regnell & Brinkkemper, 2005). Maurice, Ruhe, Saliu, and Ngo-The (2006) stress the that it is critical to remember that the requirements are due to change, but according to Aurum and Wohlin (2005) it is unclear in what way they will change. Thus, software companies need to be able to quickly take action to varying market requirements, while keeping their value proposition in mind (Barney et al., 2008). However, software does not need to be better than meeting users' needs, acknowledging that servicing is costly (Shaw, 2002). Shaw (2002) argues that users can live with a certain level of accidental problems.

#### ***The Main Objective Is To Release the Right Product at the Right Time***

According to several authors, the main objective in market-driven development is to release the right product at the right time (Carlshamre, 2002; Regnell & Brinkkemper, 2005). This imposes a certain schedule constraint and a larger emphasis on release planning (Karlsson et al., 2007). The time-to-market aspect is recognized by Rajlich and Bennett (2000) as problematic, they see that managers have a tendency towards launching the software quickly to regain the investment and win market shares, since initial development requires high investments. However, Barney et al. (2008) also stress the fact that, in order to succeed, the product's quality level should be superior to that of competitors' products.

Release planning is an important factor since in market-driven development there are several releases, and the releases are focused on evolution rather than maintenance, compared to the customer-specific development where one major release are followed by a maintenance phase (Regnell & Brinkkemper, 2005). Rajlich and Bennett (2000) explain the evolution stage as the stage where the software evolves and new features are added, and the servicing stage is focused on service of existing features.

Carlshamre (2002) stresses that release planning in the market-driven context is when development meets the market perspective, controlling what features and quality are released to what customer at what time. Carlshamre (2002) points out further differences between release planning in market driven requirements engineering and customer-specific, in a market driven environment there is a fixed release date, often a fixed amount of available resources and the planner has the assignment to choose a set of requirements to implement (Carlshamre, 2002). Furthermore, Carlshamre (2002) points out that in a market-driven development environment with millions of consumers, the planner may have a more absolute decision-role and the decision-making might seem easy, but decisions are less based on consumer information, since it is harder to collect.

### ***Success Is Measured In High Sales Figures and Positive Product Reviews***

Regnell and Brinkkemper (2005) point out the success factors of market-driven requirements engineering as high sales figures and positive product reviews, as opposed to customer-specific where success is measured in customer satisfaction and user acceptance.

Thus, the reader should recognize that market-driven requirements engineering is different from the customer-specific situation in the following ways:

- A constant flow of changing requirements are arriving from a big group of customers
- Time-to-market is important and impose certain problems regarding schedule constraints and release planning
- The success is measured in product sales and recommendations

### 3.3.2 Activities in the Requirements Engineering Process

Basic concepts in requirements engineering are requirement elicitation, requirement specification, requirement validation and prioritization. In market-driven requirements engineering the concepts of release planning and market analysis are added (Regnell & Brinkkemper, 2005). The stages examined at the Company are described in-depth and are shown in figure 12.

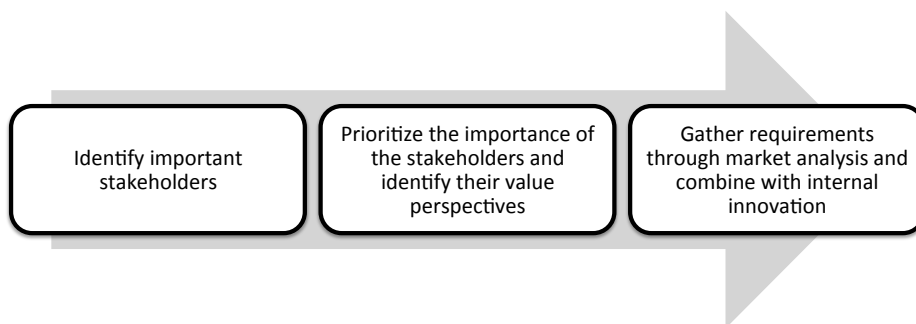


Figure 12. The three main activities in requirements engineering.

#### ***Elicitation of Requirements***

When a system is built it is important to include stakeholder requirements, and therefore the stakeholder needs and desires need to be clarified and understood (Glinz & Wieringa, 2007). The stakeholders' value propositions are often different (Jain & Boehm, 2005). The first phase in elicitation is to identify the important stakeholders. The people and organizations searched for are: users, management, development, people who have a financial interest in the system (buyers or sellers), regulators and people who are negatively affected by the system (Glinz & Wieringa, 2007; Boehm & Jain, 2005). Several authors claim that even if the identification of stakeholders and their needs and desires is seen as a critical stage in the research area, it is poorly conducted in the software industry (Pressman, 1997). Pacheco and Garcia (2012) have found two reasons for this: developers take for granted that the only stakeholders are the direct users, clients and the development team, and since it can be cumbersome to elicit all stakeholders' needs, they are eliminated or subsidized with requirements from sources that are easily accessible.

Thereafter, the importance of these stakeholders needs to be prioritized (Glinz and Wieringa, 2007). Boehm and Jain (2005) take this one step further and mean that

the value proposition of the identified stakeholders needs to be understood. Elicitation is an important phase in market-driven requirements engineering, and focuses on gathering requirements from a market analysis in combination with idea generation by employees, created from the opportunities of new technology (Regnell and Brinkkemper, 2005).

The elicitation affects the remainder of the process negatively if “high quality” requirements are not chosen and all critical stakeholders are involved. Further on in the process there are less qualitative requirements to prioritize, finally affecting the final product’s value if not “high quality” requirements are input (Regnell and Brinkkemper, 2005; Davis et al., 2006).

In summary, elicitation is important. All relevant stakeholders need to be given a chance to give their input. The input needs to be balanced so that all value perspectives are put forward in the process, otherwise the output will be negatively affected.

### ***Prioritization of Requirements and Release Planning***

The core aim in the subsequent activity is to implement the most prioritized requirements from stakeholders (Ruhe, Eberlein, & Pfahl, 2003), within the frames of time, cost and quality (Barney et al., 2008). Additionally, Carlshamre (2002) states that release planning is the activity when the requirements for a certain release is selected. The ranking procedure has a major affect on the economic value of the software, determining what is added in the final product (Achimugu, Selamat, Ibrahim, & Mahrin, 2014).

Achimugu et al. (2014) point out that to have an efficient and successful requirements engineering process there needs to be an explicit ranking of requirements. Moreover, Berander, Khan, and Lehtola (2006) mean that release planning and prioritization are important areas due to problems in software development, such as inadequate budgeting and limited resources, since not all requirements need to be addressed simultaneously.

After the second step of prioritization and resource estimation, requirements are selected for a certain release (Carlshamre, 2002). It may seem fairly easy to choose the requirements from a predefined list, and the final number of working hours

needed correspond with the available resources (Carlshamre, 2002). Nevertheless, Carlshamre (2002) claims that additional judgment is needed since it is challenging to be aware of the current context and circumstances. Moreover, another study (Carlshamre, Sandahl, Lindvall, Regnell, & Natt och Dag, 2001) shows that since requirements often depend on each other, a low priority issue might need to be resolved in order to solve a high priority issue, making planning more complex.

Release planning should be done continuously, however, it should be more emphasized after a release, when the opportunity to follow up on market reception occurs (Carlshamre, 2002). Furthermore, it should be aligned with the overall strategic product planning, meet consumer needs and track changes in and demands from the market (Carlshamre, 2002).

In conclusion, release planning and the priority of requirements are difficult activities, which affect the final product.

### **3.3.3 Common Challenges in Requirements Engineering**

Hall, Beecham, and Rainer (2002) point out that problems found in the requirements engineering are founded in organizational problems, rather than technical problems. Karlsson et al. (2007) underline that they mainly see problems connected to social aspects.

#### ***Balancing Market Pull and Technology Push***

Firstly, in order stay competitive you need to create groundbreaking technology, while matching the perceived needs of users. This is explained as a balance act between the market pull and the company's technology push (Regnell and Brinkkemper, 2005). Karlsson et al. (2007) explains that engineers tend to invent their own requirements, and might favor innovation over customers' needs. El Emam and Madhavji (1995) also see this problem, stating that: "*(Developers) try to force their ideas on the users*".

#### ***Chasm Between Marketing and Development***

Karlsson et al. (2007) point out that there is communication gap between marketing and development, affecting the process negatively since requirements are sometimes collected from marketing and thereafter forwarded to development. The authors point out that there might be a communication problem since marketing



staff see requirements in one way, which does not correspond to development's perspective (Karlsson et al. 2007).

### ***Challenges in Estimating Value and Costs***

Value and cost are according to Carlshamre (2002) the two critical attributes in release planning. However, these two words are not easy to understand. Value is a collection of strategic business values for the vendor and consumer. However, the large amount of consumers can have different interpretations of value (Regnell, Höst, Natt och Dag, Beremark, & Hjelm, 2000). Furthermore, costs can be difficult to estimate. This also needs to be matched against the different knowledge areas available and needed (Carlshamre, 2002). According to Regnell and Brinkkemper (2005) it is difficult to perform realistic cost-benefit analyses, since the estimation of both costs and benefits are error prone and depend on several aspects that are hard to foresee; the market and technological advancements and what competitors have in making. However, after the product has been released evaluating decisions is possible.

Karlsson et al. (2007) discuss a different problem regarding release planning based on uncertain estimates, that the requirement prioritization is done ad hoc and the requirements are prioritized based on which organizations is loudest.

### ***Implementing and Improving Requirements Engineering***

One of the main challenges is to find approval when changing the way you work with requirement engineering, since it involves changing the employees' behavior (Karlsson et al., 2007). It takes time to do such reorganizations and to implement new guidelines (Karlsson et al., 2007).

To finalize this section table 2 summarizes the most important problems found in requirements engineering.

<b>Challenges in market-driven requirements engineering</b>
Imbalance between market pull and push
Chasm between marketing and development
Challenging to estimate value and costs
Overloaded requirements management

Table 2. Problems in requirements engineering.

### 3.4 Summary of Theoretical Framework

Software product management aligns the three value perspectives with a life cycle perspective. The role of the software product manager is important in order for this aim to be fulfilled. The frameworks used emphasize the involvement of different stakeholders at different times and explain the main activities in software product management. The main problems caused by insufficient software product management are; *unsuccessful products throughout the lifecycle, a feature collection is produced instead of a whole product concept, delays, overruns, scope creep, rework, a changed focus during the lifecycle, slow decision-making and individuals taking ad-hoc decisions and requirements are not mapped to value creation and business cases.*

A robust requirement engineering process is efficient, transparent, involves all necessary stakeholders and aligns the market needs, the product definition and the business goals. Market-driven requirements engineering differs from customer-specific requirements engineering: a constant flow of requirements from a big group of customers, release planning is an important factor constraining schedule and resources, product sales is used to measure success. Elicitation is gathering all requirements from stakeholders and consists of both idea generation and market analysis in a market-driven context. Release planning and the prioritization of requirements are difficult activities affecting the final product significantly. The main problems in requirement engineering are: *imbalance between market pull and push, chasm between marketing and development, organizational instability and market turbulence, cost-value-estimation, overloaded requirements management and implementing and improving requirements engineering.*

The theoretical framework is a guide throughout the remainder of the thesis and helps us to further analyze the examined processes and problem areas.

## 4 Case Study Findings

---

*This chapter presents the findings of the conducted case study. The content is a compilation of the results of interviews and observations. The gathered material was structured and reviewed in order to reveal patterns and to connect findings to the theoretical framework. We have chosen to present a selection of the empirical findings that we find most relevant for fulfilling the purpose of the study.*

---

### 4.1 Introduction

To facilitate reading the chapter starts out with a general description and then goes further into detail as it progresses. Firstly, the Company context is described, that is, the smartphone industry. Then, we describe our interpretation of the processes examined at the Company. The processes are described, connecting concepts to those described in previous research. There are two types of development processes at the Company; both are illustrated with a picture, which was constructed based on findings in the interviews. Firstly, the process of how new functionality is developed and implemented is described. Secondly, the quality improvement process is explained.

The final section presents and analyzes the problem areas we have identified at the Company, with regards to relevant theories presented in the theoretical and the context that the Company operates in. Each problem area is summarized with an analysis regarding our reflections on the issue at hand, and concluded with key takeaways connected to the problem area.

### 4.2 The Company Context

In order to fully understand the Company's situation, it is important to first discuss the smartphone industry. The industry centers on embedded products (see Kittlaus & Clough, 2009) in a market-driven environment (see Karlsson et al., 2007). Moreover, the industry has gone through several fundamental changes in recent years due to the breakthrough of the smartphone.

The smartphone industry is highly competitive and has been faced with disruptive changes since the introduction and breakthrough of the smartphone. The increasing number of interconnections between companies producing products together that form an ecosystem characterizes the industry (Hacklin et al., 2013). Moreover, there has been an inevitable increased focus on software, and consumers purchase their smartphone more based on software than hardware nowadays (Hartmann et al., 2012). The increased focus on software has also entailed the possibility of changing the user experience throughout the lifecycle by using updates. We have found that the industry in general, and the case company in particular, traditionally has a considerable focus on hardware. However, the recent changes have altered how value is created.

Furthermore, as smartphones are products where software is embedded this adds to the complexity of the product value proposition. The software cannot be sold separate from the hardware, and vice versa. Consequently, it is difficult to manage the product as a whole.

### 4.3 The Interviewees

In this section, we introduce which organization the interviewees belong to (Table 3). This is presented in order to provide an overall picture of the organizations that were chosen to represent the stakeholders and decision-makers in the examined processes. For reasons of confidentiality and personal integrity, we have used fictive names for both organizations and interviewees. However, the names of the organizations gives a picture of what their responsibilities are. It shows the width of the different organizations we have interviewed during our stay. As mentioned in the methodology chapter we have interviewed employees in a decision-making role to envisage how they make decisions.

Name	Organization
Nick	Experience Planning
John	Customer Service
Paul	Development
Jacob	Scope management
Matthew	Development
Harry	Technical Sales

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

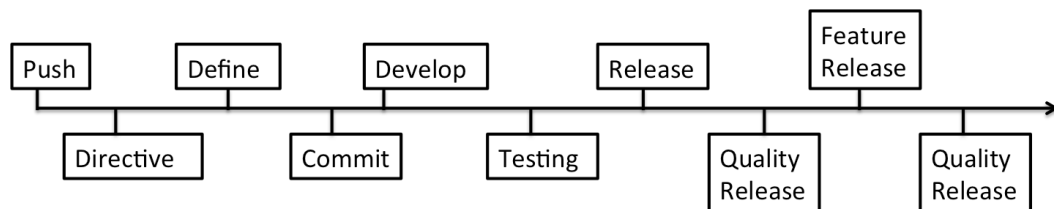
<b>Ron</b>	Technical Sales
<b>Marcus</b>	Architect
<b>Rob</b>	Experience Planning
<b>Per</b>	Usability
<b>Michael</b>	Development
<b>Stefan</b>	Usability
<b>Andrew</b>	Experience Planning

**Table 3.** Summary of the persons interviewed and their organization.

## 4.4 Overview of Software Development Processes at the Company

This section begins with an overview of the software development process (Figure 13), to explain the context of the two processes that were identified at the Company. The two process descriptions are our perception of the two development processes of new functionality and quality improvement, based on the interviews and observations we have conducted at the Company.

### 4.4.1 The Software Development Process



**Figure 13.** The software development process at the Company.

New software development at the Company starts with either a notice from their platform provider that they will release a new version of the operating system, or that the Company is to release a new lead device in need of a new version of the software. The Company releases several software versions, which are then evolved. This is typical for a market-driven context, there are several releases with an evolutionary characteristic (Regnell & Brinkkemper, 2005). The software is implemented into the lead device, and as an update to selected older devices.

Initially, the planners create a directive for the release, which describes what it should focus on. Organizations that influence the directive and provide input to the

planner are Technology, Experience Planning, Business Partner Development and operators through their two channels at the company: Operator Requirement Manager and Technical Sales. These organizations are described in the process descriptions.

The next step in the process is to define what the software should include; the legacy from previous products including new features depending on new technologies constitute input to the software release. The involved organizations will thereafter commit to develop the software for a specific product. Development then starts developing the software. There are two formal development processes, one for applications and services, and one for the platform, the difference between the two is described below.

The responsible development team then develops the software release and when it is finished it undergoes testing within the company. Thereafter it is released to the operators for a first period of testing. The operators test it and give feedback regarding issues, which are reported back to development. After the first test period, there is a final test period, where the operators finalize testing. Before the products are launched, the involved organizations sign off that the software is good enough to launch. Other partner companies need to accept the software before it is launched to the market as well. After the product is launched there are several updates. The updates consist of new functionality- as well as quality improvements. There are two kinds of releases; one focused on new features, called Feature Release, and one for quality improvements, called Quality Release. A Quality Release usually follows the Feature Release, since there are always defects and bugs that have not been addressed during initial development.

#### **4.4.2 New Functionality Process**

New functionality can be everything from new applications, to better wifi-connection. According to Rajlich and Bennett (2000) this stage in software development is called evolution. The following description is our perception of how the process has been explained to us during the interviews. We describe how the process works from an identified stakeholder need to a final feature. We do not describe the development stage and the release to the market since these are not included in the requirement engineering process. The process seems linear on

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

paper, but we found that it is more iterative and ad hoc than the picture shows. The process is described in detail in the text and can be seen in Figure 14.

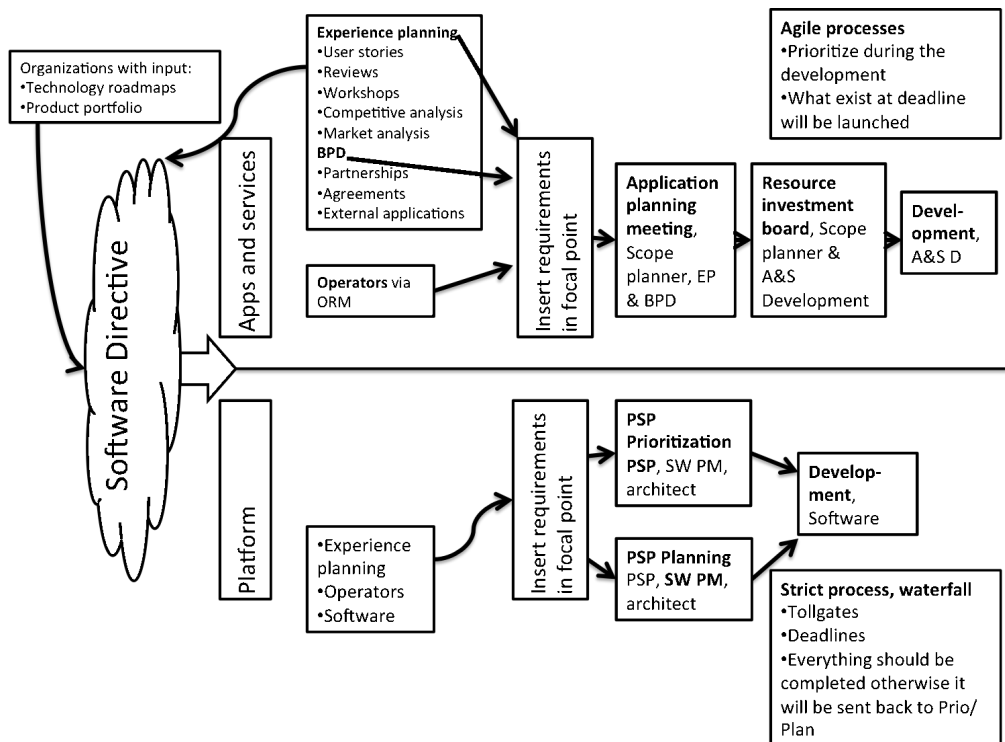


Figure 14. The new functionality process.

The planner begins by creating a directive for the next feature release essentially doing a roadmap for what to include in the release (see van der Weerd et al., 2006). The directive influences the applications and services flow as well as the platform flow. It is not static, but dynamic and is updated during the process. A planner is responsible for making the directive and will be influenced by several other organizations during the creation. Prior to the creation of the directive, portfolio management decides on the product portfolio. Such decisions are mainly concerned with which software to include in which project. This is the phase of portfolio management in software product management (van de Weerd et al., 2006).

The organizations involved in developing new functionality are Experience Planning, Product Portfolio Planning, Technology and Business Partnership Development.

## Problem Areas in Creating Valuable Smartphone Software - A Software Product Management Case Study

---

*Experience Planning* is responsible for making user stories for new features, applications and services, and they are also product owners for specific applications and services. They employ user cases, workshops with organizations such as Technology and Companion Products, market analyses and competitive analyses. They participate in making the directive by telling the planner what their current focus is. *Business Partnership Development* will also give input, they are responsible for partnerships with external applications not produced in-house, and are the owner of the agreements.

Further on, the process is divided into two parts depending if the feature is in an application or a feature is closer related to the platform. Application and Service Development is the executing organization if the feature is related to an application, and Software Development is the corresponding organization in the platform part.

A tool called Focal Points is used to gather all requirements from stakeholders in both processes. This is referred to as requirement elicitation in literature (Regnell & Brinkkemper, 2005) (Glinz and Weiringa, 2007).

The stakeholders are the same in both flows, consisting of Experience Planning, Business Partner Development and operators. Experience Planning and Business Partner Development write their requirements themselves in Focal Point, and the operators are represented by an in house organization, Operator Requirement Managers. Experience Planning and Operator Requirement Manager are the stakeholders who produce the most requirements.

### ***Applications & Services Features***

The application and service flow starts with an Application Planning Meeting, which the application planner owns. In this meeting several organizations are represented. Experience Planning present their suggestions for applications and services, together with Business Partnership Development, who present the external application perspective. The aim is to prioritize requirements in Focal Points and to decide whether to invest in the application project or not. They also sometimes decide which products should have which application. Prioritizing requirements is a common stage in requirement engineering (Achimugu et al., 2014).

After the Application Planning Meeting, another meeting called Resource Investment Board is held. The main objective is to allocate resources within the



development organization, to different applications and services. Resource allocation is also a stage in requirement engineering (Carlshamre, 2002). The Application and Service Development organization own the meeting. They are represented together with the Application Scope Planner. Essentially, Application and Service Development will accept or decline the conditions that were decided on during the planning meeting.

### **Platform Features**

In the platform flow there is only one meeting, where prioritization and planning is combined in order to be more efficient. The scope planner owns the prioritization, and prioritizes the features. The software product manager owns planning. There exists one meeting for each hardware device. They prioritize requirements from stakeholders, in alignment with the purpose of the release and strategy. The planner communicates how he intends to prioritize and the project manager gives feedback as for how he plans to move forward with is operationally.

Both in the Application and Service flow and in the platform flow there can be a reevaluation after the development organization estimates the resources. If the costs are too high compared to the estimated business value, the Scope Planner needs to reconsider the investment.

The activities described in this paragraph are concerned with what is referred to as release planning and prioritization in literature (Achimugu et al., 2014; Carlshamre, 2002; Ruhe et al., 2003), which is prioritizing requirements based on their impact and selecting the requirements to include in the next release.

### **4.4.3 The Quality Improvement Process**

Quality improvements are used to correct bugs, issues and defects found after the software has been released to consumers. The purpose is to improve software quality over time, either by correcting issues that have not yet been corrected at the point of release or new issues that occur due to changing circumstances. This is referred to as servicing by Rajlich and Bennett (2000). It is a reactive activity by definition, aiming at enhancing the quality of the software that has been distributed. A typical quality improvement release solves technical issues with the software, such as interrupted sessions, problems connecting to the network or Wi-Fi or Bluetooth functionality issues. In relation to the activities in Figure 13 the issues are when the

product is launched, but the fixes could be included in to both Feature Releases and Quality Releases.

Quality improvement release development is schematically illustrated in Figure 15 below. The reader should take notice that the process is more iterative and ad hoc than it looks, and it is not a formalized process per se.

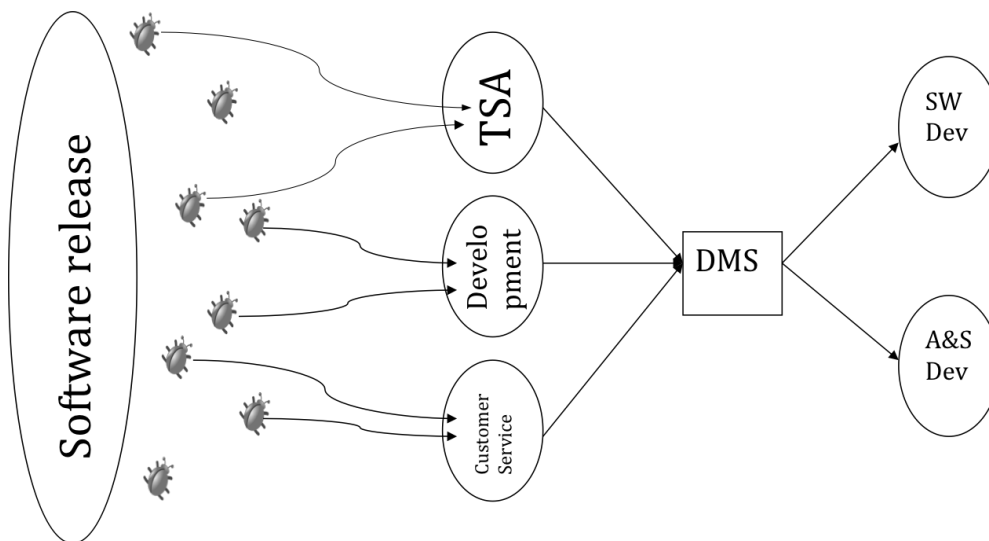


Figure 15. The quality improvement process.

### ***Discovering and Reporting issues***

Many defects have already been discovered at the time of a software release, through both internal testing and operator testing. Nevertheless, there are always undiscovered issues that surface after the software has been launched. Such defects are discovered by three main entities.

Technical Sales receive defect reports from the operators, which could be based on what they find in their test processes or defects that are reported through the operator's different channels, such as call centers. Technical Sales is the organization responsible for technical sell-in to operators. They are also responsible for releasing updates together with the operators, and the activities that follow, such as collecting issues from operators and they also negotiate which issues need to be prioritized. Some defects are discovered by Development as a result of their usage and testing.

Finally, Customer Service reports defects that are based on the results of scanning social media, blogs and forums, as well as call center and repair center reports. Customer Service collects plenty of information from the servicing touch-points they have with consumers. The defects and bugs that they discover are reported as “issues” in a system called DMS, where they are described and assigned to a development organization responsible for solving them.

### ***Prioritizing Issues***

Issues in DMS are flagged with an initial priority based on the assessed severity of the defect. A priority 1 issue must be fixed before the software is released, priority 2 issues must be fixed in the next Quality Release, and priority 3 issues are reported but not considered crucial. These priority measures are available for all roles with access to DMS. After issues have been reported in DMS, they are addressed by the organization concerned. They set an internal priority, and solve the issue in due time.

Usually, the development teams themselves prioritize their issues. The teams are divided functionally based on their specific area of expertise, for example Wi-Fi, Bluetooth, telephony, etc. These so called enabler teams are responsible for issues connected to their particular area, and prioritize amongst their issues and tasks at an operational level, unless something is over-ridden by the software project. The software project might prioritize some issues in terms of their severity or impact, as well as the available resources. However, the software project manager says that he avoids interfering unless there are conflicts with stakeholder organizations.

The Company uses the so called User Experience quality forum, in which issues reported in DMS that are connected to user experience and usability are reviewed on a regular basis. The purpose is to lift the most important issues. This forum consists of representatives from Software Development, the User Experience and Planning organization and Customer Service. The representatives from software development are from the teams, and are included based on the issues found.

## 4.5 Problem Areas at the Case Company

In this section we introduce the areas where we see some problems in the Company. They are based on what the interviewees have expressed as problems, and are analyzed using the theoretical framework including relevant theories. Under each area we explain what the problem area is, describe how it is challenging for the Company to overcome it, and finalize each section with a short description how they could overcome it. The section contains an introduction with the empirical evidence, a comparison with the relevant theory and lastly we discuss the findings in a short section

The problem areas presented below are: *Lack of a common vision, Non-existing software product management role, Unpredictable and non-transparent requirement prioritization, Consumer awareness, but no consumer focus, Imbalance between idea generation and market analysis in elicitation activities and Operators are seen as the most important stakeholder.* A table has been compiled to show the main organizations involved in the processes (table 4). Furthermore, the table shows their main activities and we have based on our interviews assessed on how much impact they have in the decision making in the processes described above.

Stakeholders	Represented by in house organization	Main activities	Impact
<b>Top management</b>	Management	Roadmapping, portfolio planning and operational decision	High, when expressed
<b>Research and development</b>	Experience Planning and Technology	Technology roadmaps, competitive analysis, usability, user interface	High
<b>Development</b>	Operator Requirement Manager, Application and Service Development, Software Project Leader, and Project and Software Development	Develop software, resource owners, interpret requirements	High
<b>Sales</b>	Technical Sales and Sales companies	Represent operators in house, negotiate with them, collect feedback from them	Medium
<b>Service and distribution</b>	Technical Sales	Preparing launch and reporting bug fixes	Medium

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

<b>Support</b>	Customer Service	Collecting information from touchpoints with consumers	Low
<b>The market</b>	In house representation: Technology, Experience Planning, Technical Sales	Competitive analysis	Medium
<b>Partners and Suppliers</b>	Operators	Input to requirement engineering process	High
<b>Customers</b>	In house representation: Customer Service, Usability and Technical Sales	Input to requirement engineering process	Low

**Table 4.** Table showing main stakeholders in the theoretical framework, compared to the involved stakeholders at the Company, their main activities, the name of the organization representing the stakeholder in the Company and how much impact they have

#### 4.5.1 Lack of a Common Vision

The first exhibited problem area that we have seen is a lack of a communicated vision. An interviewee from Technical Sales underlines that the vision that they have is not really their own:

*Currently, we ask the operators what they think will be important in six months or a year. I think that it is us who should make that decision and we should push the agenda that we believe in.*

According to another interviewee at Experience Planning, the reason behind the difficulties in planning and prioritizing requirements is the lack of a common vision for the software:

*The company does not have a clear common vision for what we want to achieve with our software suite.*

He also says that they do not have a clear prioritization of what the Company wants to offer consumers. He believes they should be working more with these issues. According to the interviewee the vision needs to be clearer in order for the Company to effectively and efficiently prioritize requirements and allocate resources.

## Problem Areas in Creating Valuable Smartphone Software - A Software Product Management Case Study

---

Aurum and Wohlin (2007) claim that the aim with value-based requirements engineering is to align software development with consumer requirements and strategic business goals. Thus, if there is a lack of clear vision as showed in the empirical evidence, aligning the development process is difficult, resulting in a process that does not generate business value. One main problem the Company has is to create products that succeed in the market place. The lack of a common vision can be seen as one of the problematic areas contributing to this problem.

Ebert (2009) takes another point of view on the problem with the lack of a common vision, it can be seen as one of the root causes of software development problems such as wrong content, rework and scope creep. Thus, creating a common vision for software can be seen as one of the important areas for the Company to start working more efficient in. If they create a common software vision, aligning their overall business goals with market requirements and communicating it thoroughly to stakeholders a more valuable product can be the result of it, if listened to Ebert (2009). In a later article Ebert and Brinkkemper (2014) state that a lack of strategy and roadmaps is common when a software product management is missing. As we describe further on, a software product management role is also missing at the Company making it a remedy to implement a software product management agenda to handle the problem and make the software product manager responsible for creating the vision.

### ***The Software Directive Is an Attempt To Create a Common Vision***

Even if some interviewees claim that there is a lack of vision we have seen an attempt to create one, on the strategic and tactical level the product scope planner develops a software directive. The purpose of the directive is to create a vision for what the software is supposed to focus on, aligning software to higher-level strategic objectives. Thus, the directive is constructed early on, and aims at providing guidelines and governance for both applications and platform features. The product scope planner describes the directive as follows:

*You could say that the product scope planer is responsible for creating a directive at the early stages of planning a certain software release. This should a guideline for stakeholders, so that they know what I, as the product owner, want.*

## Problem Areas in Creating Valuable Smartphone Software - A Software Product Management Case Study

---

Higher-level management approves the directive, globally or locally, depending on the scope of the software release.

The product scope planner states that a directive is always constructed for the initial software releases as well as for most of the early updates but not for all software releases. The point is to anchor the idea with all stakeholders, or at least to communicate it to them. The product scope planner stresses that it should not be a surprise what he prioritizes. Subsequently, the directive should provide guidance as to what feature requests to send in. It is unclear who the stakeholders are in this process, also to the product scope planner himself. Also, interviews with product scope planner and operators requirements managers showed that the directive is not set in stone at its inception, but it is the result of a very iterative process, and the directive changes over time.

As Keller (1995) points out stakeholders define value from different points of views, and Aurum and Wohlin (2007) thus emphasize the importance to involve several stakeholders in the process to get various perspectives of product, project and business value. Since the product scope planner is not fully aware of the stakeholders involved in the process of making the vision, and the stakeholder interviewed does not feel that it is communicated to them, we cannot see that all the important perspectives are taken into account when the directive is made. Therefore, it will be difficult for the Company to release software that is valuable to consumers. Hence the consumers will probably buy other products from companies making more valuable products.

### ***Discussion***

It is reasonable to assume that the iterative process of creating the directive makes communicating it difficult, as it may change from one week to another. Furthermore, as it changes, you risk that stakeholder's act on old information. Even though the software directive is created and supposedly communicated to the relevant stakeholders, it has been lifted in the interviews that the prioritization is unclear. This is probably the result of the directive not being properly communicated or unclear. Whether this depends on its changing nature or that it is communicated to the right stakeholders, or that it is not being followed by development remains fuzzy. If the software directive is created, and people still claim that there is no software vision, something is obviously not right.

Even if some of the interviewees claim that there is a lack of a common vision we have seen empirical evidence that there is an attempt to create a common vision at the Company. However, since we do not see an established understanding of this document we conclude that there is a lack of a common vision, and additionally, that the vision that is implemented is inadequate. The Company has a big challenge to overcome this. It needs to start working towards integrating a software vision and making it visible to stakeholders, telling how the software should work and what the main values through it are. To be able to do this all the relevant stakeholders need to be identified, both within and outside of the Company, their thoughts should be considered when the directive is made.

#### **4.5.2 Non-existing Software Product Management Role**

In examining how the Company's problems relate to software product management, it is relevant to examine the software product manager role, as it is central to the research area. The following paragraphs compares the software product manager role described in literature to the roles found at the Company.

First of all, it can be concluded that the Company does not have a role that they explicitly refer to as a software product manager. However, there are roles that are similar to the software product role described in literature.

##### ***Technical Aspects Handled by the Scope Planners***

The product scope planner identifies himself as the product owner. At the application level there are also assigned experience planners, who are responsible for the development and maintenance of one application. An experience planner refers to this as a kind of product ownership:

*More or less all [applications] that we develop or maintain should have an assigned experience planner, who is responsible for roadmapping and execution, and this person is the product owner.*

Moreover, the quality improvement releases are not managed by either of the scope planners but by the software project.



In accordance with software product management literature (Fricker, 2012; Ebert, 2007), in this sense the product scope planner acts as the software product manager, as he considers the lifecycle of the software platform, and evaluates which functions to retain, evolve or terminate during prioritization meetings. However, there are different scope planners for each flow. This could be interpreted as the different flows being regarded as different value propositions, that is, different products.

The product scope planner is responsible for managing requirements and defining the software releases. These rather technical aspects are consistent with the descriptions of the software product manager in literature, such as van der Weerd et al. (2006). The authors defines the software product manager responsibilities as managing requirements, defining releases and defining products in a context comprising of several internal and external stakeholder involvements.

### ***The Scope Planner Does Not Have a Life-cycle Perspective***

The implementation of the strategy plans is the creation of the software directive and the prioritization and planning of features and resources.

According to Fricker (2012) and Ebert (2007), the software product manager implements strategy plans by coordinating departments such as development, marketing, sales, distribution, service and support, which he also does in the Company creating the software directive. The directive being created by the product scope planner at development highlights the software product management aspects of the scope planner role. The directive aiming at guiding both the application and the platform flow further highlights that the product scope planner role contains important aspects of the software product manager role.

He continues to be responsible for scope planning at the development side. The scope planner says on the prioritization meeting:

*This is my meeting, as the product scope planner, I own this meeting. We have one for each product, where I prioritize the features for the product at hand.*

However, he is not responsible for prioritizing requirements and scope changes at the application side, and therefore his responsibility does not cover the full life cycle.

He does not govern the entire process, nor is he held responsible for the product at an aggregated level. This applies to both the application flow and the platform flow.

### ***No Role Spanning Multiple Functions***

First of all, the software update process itself is quite fragmented. It is divided between different functions at different points in time. One of the interviews illustrated the process by saying:

*The software update process is very long, it starts by someone submitting a case to software, and it ends by something coming out at the other end, but it happens a hundred things in between.*

Additionally, the experience planner claims that his organization owns the user experience, even for the platform flow. According to him, they are involved in the end of the process and review the results and sign off on them:

*We are not really included in that process, but we still have some kind of veto over the user experience.*

When asked if there is a formal ownership of the process, all interviewees except one claimed that there is no formal ownership and that the process is rather a collection of sub-processes owned by the different executing organizations in each step. The interviewee that claimed that there is formal ownership over the process was himself previously the formal process owner.

One of the interviewees from Technical Sales compared the process to a relay, where the baton is handed over from one organization to the next in line.

Ebert (2007) contrasts the unclear ownership aspect by claiming that the lack of formal authority for the product owner and unclear ownership are significant obstacles in achieving product success throughout the lifecycle.

Fricker (2012) and Ebert (2007) summarize the role of the software product manager as the interface between business and technology functions. Here, there is a significant difference in the role of the product scope planner and the software product manager described in literature.

## Problem Areas in Creating Valuable Smartphone Software - A Software Product Management Case Study

---

The product scope planner is first and foremost a technical role, and the interface to business functions is not apparent. According to literature (Ebert & Brinkkemper, 2014; Barney et al., 2008) software product management aims at aligning the product, project and business perspectives and that the software product manager must have some authority in all these three functions. It is clear that there currently is no such role at the Company when it comes to software. There is some alignment between the product and project, however the business perspective is underrepresented.

According to the software project manager he is in fact the one who has the most authority when it comes to what features are included:

*The one who has the most impact is the software project manager.*

The software project manager is responsible for distributing work and resources; therefore it is reasonable to assume that he will be instrumental in deciding what actually gets done.

This underlines that the scope planner does not have the authority described in literature. Also, since there are very different ways of working with the applications and the platform, and the product scope planner is not responsible for following up the directive at the application side, he does not have the authority over those functions either. Since no one has clear authority over several functions, aligning the different organizations around one value proposition becomes difficult.

However, it is reasonable to believe that all software aspects contribute to the user experience, as it concerns not only the user interface, but also how everything works together. Therefore, the user experience should be something that is better coordinated between organizations.

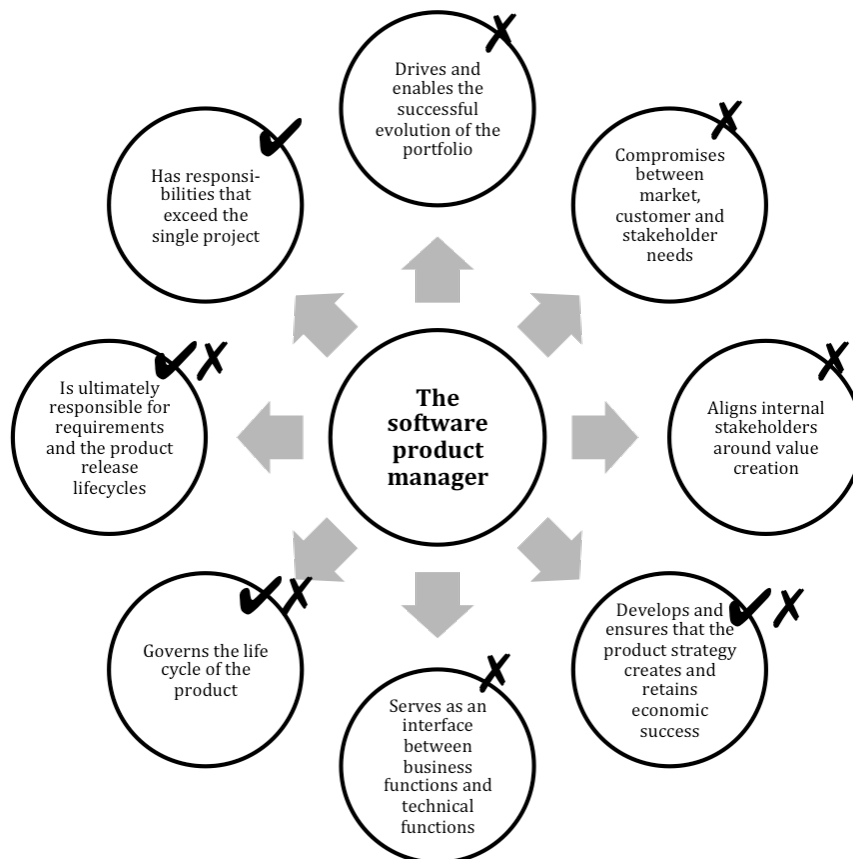
### **Discussion**

There is no defined software product manager role within the Company. There are, as has been argued above, similar roles. However, we conclude that several important aspects are missing. We summarize the software product manager aspects that are, and those that are not, present in the roles described at the

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

Company in Figure 16 below. The figure is based on the role described in the theoretical framework.



**Figure 16. The software product manager aspects that we have found and not found within the Company (Fricker, 2012; Bekkers et al., 2010; Ebert, 2007; Van De Weerd et al., 2006).**

Firstly, we can conclude that no one has authority over the entire life cycle. The product scope planner creates the software directive, but has no way of controlling that it is followed, due to the different flows. Secondly, there is no role that serves as the interface between business functions and technical aspects. Furthermore, the roles found in the Company do not align stakeholders, and definitely not value creation. The product scope planner is not involved in portfolio decisions and therefore he could not possibly drive and enable the successful evolution of the product portfolio.

There are some aspects of the software product manager that are present to some extent. The product scope planner creates the software directive, which is most similar to the product strategy. But, the product scope planner does not follow up on it, and has limited ability to do so, and therefore we argue that it is not possible to ensure that it creates and retains economic value. The product scope planner does to some extent govern the lifecycle, but only for the platform. Subsequently, we deduce that the product scope planner cannot govern the entire lifecycle. As for the ultimate responsibility for requirements, this is not completely the case either, also due to the two development flows. The product scope planner does, however, have responsibilities that exceed the single product.

One needs to consider the fact that the Company works with embedded products. As a result, product ownership is complex. It is also not a single software product, but it consists of different kinds of software. Consequently, many stakeholders have a different perspective of what the product actually is, where some claim that it is the hardware, and some claim that it is the collection of all aspects.

It is likely that the product complexity makes it difficult to have a software product manager as the role is described in literature. However, the activities of the software product manager remain important from a value creation perspective. Therefore, it could be useful to implement a core team that represents the different product aspects that are found within the company, that is, both software and hardware aspects. Their responsibility could be to govern the lifecycle of the entire product, thus managing the entire value proposition, uniting different aspects into one complete message.

### **4.5.3 Unpredictable and Non-transparent Requirement Prioritization**

The prioritization activity at the Company is unpredictable and non-transparent according to interviewees, making them questioning the activity. This section explains and reviews this problem area.

Several of the interviewees have brought forward that the priority of requirements is unclear. An architect says that this is a general problem:

*Requirement definitions are unclear. Generally requirement definitions are unclear here.*

Another interviewee from Technical Sales agrees and says:

*It is unclear to us why some things and features are put into some products but left out from others.*

The two development processes also have different way of prioritizing making the prioritization more complex.

### ***New Functionality Process with Two Competing Prioritization Processes***

One interviewee means that there are two processes for prioritization in new functionality releases, first of all the formal process where the planner prioritizes in the priority meetings, while the development teams have their own backlog with requirements where they make their own prioritizations. The planner needs to justify and motivate why requirements should be included, i.e. there needs to be a business value. However, the backlog prioritization is more managed intuition. This implies that there is no one that has an overview of the requirements. The architect explains it like this:

*Each team has a backlog, which is owned by the product owner. It is he who owns the priority ... It's a bit vague, since the requirement definition is made by various organizations.*

There are two different processes parallel to each other, and therefore it might be hard for stakeholders to understand who makes decisions about the requirements.

The empirical evidence shown above highlights the problems connected with non-standardized processes in companies without software product management, which is well in line with what Ebert and Brinkkemper (2014) describe. In the new functionality release there is one formal prioritizing process, but also an informal one. There is no formal process spanning all of the prioritization. A challenge to overcome arising from these non-standardized processes is that employees tend to make ad-hoc decisions according to Ebert and Brinkkemper (2014), which may be

the case in the Company as well. To reduce ad-hoc decisions there is a need to formalize the processes so it becomes evident who is responsible for prioritizing and making the decisions.

The empirical evidence also shows that developers at the software departments are sometimes responsible for the prioritization. This can be problematic considering that Aurum and Wohlin (2007) claim that software developers operate in a value neutral world and are not used to take value perspectives into consideration. As the interviewee stated above the development teams prioritize based on intuition, which could be compared to a value neutral prioritization by for example not taking into consideration what consumers want. The intuitive decision-making is also closely linked to the problem with ad hoc decision making. To correct it, the Company can make sure that development teams are familiar with all the perspectives or alternatively give all the prioritization power to the product scope planner who should make sure he has knowledge of all the stakeholders value perspectives.

Referring to Regnell and Brinkkemper (2005) and Wohlin and Aurum (2005) it can be challenged whether the Company selects requirements in a way that creates value since the prioritization is unclear. There is a lack of motivation for what is added in the products shown in the empirical evidence. This could be a communication problem, meaning that the Company is not able to communicate to stakeholders why certain requirements are chosen. Furthermore, according to Wohlin and Aurum (2005) there is a need to transfer information and knowledge from and to stakeholders in order to achieve a process where value is created. A transfer of information and knowledge cannot be seen either in the empirical evidence, since the interviewees are questioning the decisions. There is a need to increase the knowledge and information transfer in the prioritization process to lower the suspicion against the prioritization process.

### ***Quality Improvement Process where the Loudest Organization Will Be Most Considered***

According to our interview with the software project leader the overall decision to make a quality release is made by the project leader. However, he explains that in the quality improvement process the teams can themselves prioritize. According to the project leader the software project manager is only involved if there are any

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

problems or there is a need to mediate. When prioritizing in this phase, the focus is the technical aspects, ensuring that the system works.

DMS is used for stakeholders to report issues and stakeholders also give them prioritization. In order for an organization to be heard they need to make extra effort. An interviewee from Technical Sales explains:

*We cannot do much more than push our priorities, which we do - by the coffee machine, via mail and in the system.*

The software project leader is responsible for the release planning, what requirements will be added in the different releases. However, one of the interviewees says that there are no resources available for improving the software after the first hardware product is released to the market.

As mentioned previously there is also a problem area related to allowing the developers to prioritize their own list of requirements as they are used to working in a value neutral world according to Aurum and Wohlin (2007). Key stakeholders' requirements may be missed since the developers do not have the full picture. In order for the Company to overcome this problem area there needs to be awareness among developers that stakeholders are important to listen to and that the consumer is the main focus.

In the empirical evidence there is a problem area concerning that the organization that will make most noise will be prioritized, namely the organizations having most resources. Since the Company operates within the market-driven context there are extreme amounts of requirements as the consumer base is large. In this environment Regnell and Brinkkemper (2005) points out that there may be a overflow of requirements, which may be the case here as well. If there are too many requirements it becomes increasingly difficult to prioritize, and there is a risk that developers listen to the organization that argues most loudly for its needs. According to Pacheo and Garcia (2012) it is also easy for developers in a prioritization process to listen to stakeholders that are easy to communicate with and substitute their requirements for others that are not as easy to communicate with. This might also be an explanation of why some of the stakeholders are more listened to than others. The Company needs to work actively with involving all relevant stakeholders so they



can give their requirements to the developers, not only listening to the one making most noise. This area could also benefit from the advice given to the Company above; a more formal process involving the key stakeholders would facilitate the prioritization of their requirements.

Another problem area identified at the Company is that there are no resources available for release planning and improvements after a hardware product has been released. As Carlshamre (2002) argues release planning should be based on what consumers want and the most emphasize should be put when an update has been released. However, as shown this is not the case. The Company should therefore put more effort into evaluating how the updates were received in the market and allocate resources so that this can be done. Otherwise the updates will not be valuable.

### ***Problems in Estimating Benefits and Costs***

We recognize that there is a problem at the Company in estimating benefits and costs with software updates. It finds it difficult to estimate how much a software update changes the consumers' perception of the product. One of the interviewees highlights the problem:

*It is very difficult to prove that an investment gives return in terms of more smartphones sold. Therefore you need to go on intuition as well.*

Regnell and Brinkkemper (2005) and Karlsson et al. (2007) recognize the problems connected to estimating costs and benefits in a market-driven requirement engineering context. The benefits are hard to convert to a monetary value, and it is also difficult to know how much more satisfied a consumer will be, and how this satisfaction will affect loyalty.

Shaw (2002) claims that consumers have a bigger resilience to bugs and other faults, than developers believe. Therefore this needs to be checked with in-house consumer representation when prioritizing. This is currently not done at the Company. Thereby, they risk making too many corrections or fixing the wrong ones.

In conclusion, the Company could definitely benefit by conducting cost and benefit estimations in order to create business cases that can be used to evaluate if an update should be developed or not.

### **Discussion**

As Achimugu et al. (2014) explains prioritizations have a great impact on the final product. Therefore it is important for the Company to take a look on how it does this. As explained above the prioritization process is not transparent. In the time we have been at the Company the process has been met with a lot of suspicion. This can be explained by the inconsistent way of prioritizing and different persons prioritizing over the lifecycle. The inconsistent prioritization coupled with the missing lifecycle perspective makes it improbable that prioritization is performed in a value-adding manner. Since there is evidence that the Company does not have a robust and clear priority process, it needs to work with both motivating why requirements are prioritized or not, as well as improving the selection process. This could also be connected to the lack of common vision, not having a guide when prioritizing. Software product management makes the process more predictable, so further implementation of software product management perspectives could clarify the process.

#### **4.5.4 Consumer Awareness, But No Consumer Focus**

This section discusses how the Company works with incorporating consumer data in the processes. As is shown, there is much to be desired in terms of pushing the consumer agenda internally. However, there are some indications that there could be a shift in focus.

The Company identifies itself as a consumer oriented company; it is explicitly expressed from top management that consumers are in fact the primary customers, regardless of the sales channels. However, when asked which the most important stakeholders are in the development process and in defining software releases, none of the interviewees identified the consumer as a stakeholder. One of the planners lists the stakeholders, not once mentioning the consumer:

*There is a defined number of organizations we call stakeholders, where the operators are one of them. They are not, however, represented by Technical Sales but by the Operator Requirement Manager. They are really those who handle the technical*

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

*requirements. Experience Planning is a stakeholder. Technology is a stakeholder. Customer Service and industrialization are also stakeholders, but they have almost no requirements. Everyone is a stakeholder; there is an "other" group who can send in requirements. But the main flow comes from the Experience Planning and Operator Requirement Manager.*

During the interviews, it became clear that the Company came across some problems in implementing a consumer driven process. One of the interviewees from Technical Sales underlines the missing information regarding what the market wants.

*I believe that we are missing specific facts about what the market wants.*

As a market-driven Company, there is a need to consider the voice of the consumers. This must be done in order to generate business value, originating from product sales, and product value, originating from the customer and market requirements (Aurum & Wohlin, 2007). Not mentioning consumers as a stakeholder implies that the processes of defining and developing software releases are not centered on delivering value to the consumer. As Ebert (2009) describes, a usual problem area in software companies is that key stakeholders are not integrated. As can be seen by the empirical evidence, the Company does not think of the consumers as stakeholders. Van de Weerd et al. (2006) highlights both current and future consumers as an important source of requirements. This would therefore imply that the Company is lacking what could be considered one of the most important sources of requirements. This would thus mean that the Company is lacking an important source of requirements, maybe even the most important.

Regnell and Brinkkemper (2005) take another point of view and claim that there are extreme amounts of requirements to consider, as well as the notion that they constantly change in an unpredictable way presented by Aurum and Wohlin (2005). The Company should be very alert and integrate the consumer perspective in its development process.

### **Consumer Data Is Collected But Not Utilized**

There are organizations whose primary concern is to evaluate and compile user perspectives and facts, such as Customer Service and Usability. We can however

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

conclude that the information is seldom used as input to the software processes. Customer Service compile reports on the voice of the consumer, which is the result of gathering data from different channels. However, these reports are not formally used as a means of discovering consumer needs. An employee at Customer Service emphasizes the problem:

*But I still believe that development and Customer Service could be much better at working together and to ensure that the information we get from the market reaches them directly. I mean, a year ago development probably did not even know that we collected it.*

Furthermore, another interviewee, one of the planners, seems to think that the Customer Service reports could be better utilized and distributed in the Company, and that this feedback should be taken advantage of in order to better understand the consumer. One of the interviewees highlights the need for integrating Customer Service better in the process by saying:

*Working more with Customer Service, there are very many consumers that contact us. We should take better advantage of this feedback – what do they think and what do they complain about?*

One scope planner states that he is aware that there are reports on consumer data, and that these are communicated to development, but he points out that they have trouble properly distributing such information. Moreover, he says that in his role, he is not aware of what the consumer wants or complains about, and therefore it is difficult for him to integrate these aspects in his decision-making.

*Some surveys are made, from Customer Service; the information is given to the software projects. We also receive some information from our service centers, and there is some information coming via the operators. But I would say that overall we have some problem receiving that information. In my role, it is hard to get it; I do not know what the consumer wants or complaints about.*

It should be difficult for him to prioritize in a consumer focused way, delivering products that are perceived as valuable, when the consumer requirements do not go as far as to the prioritizing stage.

That organizations work in parallel is not unheard of, and that Customer Service collects information for the Company, which is not utilized, is a typical example of silo work. This is similar to the disconnections between departments discussed by Ebert and Brinkkemper (2014), who claim that it is common that software organizations work in silos, not considering the work of other organizations. This of course rules out the possibility of generating synergies between departments. Ebert and Brinkkemper (2014) mean that a more formalized process could help to overcome these problems. The Company could therefore formalize and give mandate to the organizations representing consumers to integrate these stakeholders in the process.

Why consumer aspects are not pushed by Software Development remains unclear. It might be that requirements are difficult to understand or that it is cumbersome in general. This confirms the problems met in the market-driven context frequently highlighted by literature (see Regnell & Brinkkemper, 2005; Karlsson et al., 2007; Regnell et al., 2000). Karlsson et al. (2007) highlights that it is hard for developers to understand consumer requirements, as they are not as explicit as developers would like. However, it may also be the result of the developers not considering the consumer as important. This would certainly explain why consumers are not mentioned as a stakeholder. Another reason for the information not being further processed could be that development does not take Customer Service seriously. The Company should focus on spreading the word that consumers should always be the top priority and since operators have their own requirements translator the same could be possible for consumers.

### ***There Are Indications of Consumer Awareness***

Despite the difficulties discovered in implementing a consumer oriented approach, there were several signs pointing in the direction that the interviewees are aware of the problems. This was primarily expressed by their reasoning around what they should be doing. One of the interviews said:

*What we want to do is to take the consumer more in to consideration. If it is an update later in the lifecycle, it should consider what the consumers thought about the initial software and what they thought about the first update, if this is the second. What problems did we experience, what did they complain about? If it is the*

*camera, then we should focus on the camera. That is what we want to do more. Now, the focus is mainly on our proposition, what we want to push, which Company values.*

This is also consistent with comments made by other interviewees on the subject of consumer focus. The fact that these ideas are circulating within the case-company is a good sign that they have at least started recognizing the problem.

The increased need for consumer centered operations is also underlined by the changed conditions for the industry, presented by Hacklin et al. (2013), where consumers now favor a market-driven ecosystem of applications rather than the operators own services, and consumers are able to choose the applications and functionalities that they want.

### **Discussion**

Since the consumer does not seem to be the focus of the value creating operations, it will be difficult to produce software that the consumers value. There are many organizations at the Company collecting important information that could be used in the improvement of the process. However, the information is not distributed for some reason. Obviously, the fact that resources are spent collecting information that is not being properly utilized is a problem in itself. The activities generate costs, but the Company does not realize the benefits of them. The information could be value adding if utilized in the processes, since it is argued in literature that it is crucial to know what the consumer wants. Thereby, we conclude that there needs to be more curiosity within the organizations responsible for prioritizing requirements regarding what the consumers' want.

Since the consumers' requirements are not included in the requirements process, their preferences will likely not be included in the end product. The Company clearly requires a shift in mind-set if they are to compete in the market-driven context. The consumer should be the main focus in all activities. We have seen important indications of an increased consumer focus, which is the first step towards achieving this shift. We must also consider that we have interviewed mostly managers and they may have a different perspective than the employees that actually develop software. Consequently, we mean that there is still work to be done in order to shift the focus of the entire organization.

#### **4.5.5 Imbalance Between Idea Generation and Market Analysis in Elicitation Activities**

This section describes the imbalance between idea generation and the marketing analysis that exist in the Company. There is a strong focus on internal idea generation, and synchronizing the ideas with market needs is problematic.

##### ***Experience Planning's Market Analyses are Incomplete***

The organization Experience Planning is according to an interviewee from the organization, responsible for doing the market analysis. However, the market analysis is more based on what competitors do, the employees' own ideas, and ideas from other organizations at the Company. There are no guidelines in the organization that specify what to include in these market analyses, and there are several other organizations gathering information that could be valuable for the experience planner. However, according to one of the interviewees this is not collected if the planner does not see the need for doing it:

*It is the planner who is ultimately responsible for gathering all the information. So it is in his decision to contact customer support or other organizations.*

This implies that the Company is disregarding valuable information and that information is sought online in forums and through other sources when Customer Service already has compiled the information needed.

The market analysis should be updated every four months, but our interviewee points out that this is not done as proactive as he wishes:

*It is not done as continuously as we hope. But it is because we lack resources now. We have more areas than we have employees. We need to do it as sprints, you need to remind the responsible for Samsung that now they have launched S5 information, I would like you to update your material and benchmark again.*

There is an evident problem in this area, where Experience Planning's market analyses are incomplete. They have a focus on what they want to promote and do not map this against the needs of consumers. In market-driven requirement engineering the previously mentioned elicitation of requirements focuses on

gathering requirements from a market analysis in combination with idea generation by employees, created from the opportunities new technology gives (Regnell and Brinkkemper, 2005). However, Regnell and Brinkkemper (2005) and Karlsson et al. (2007) identify problems in balancing market pull and technology push, where engineers may have a tendency to invent requirements instead of seeing the real need from users. Therefore, the Company could emphasize the importance of taking the consumer into consideration when doing the market analyses. We can definitely see an imbalance at the Company, since there is limited consumer market input used in the requirement engineering process.

Again we can see the problem with the unformalized process as Ebert and Brinkkemper (2014) describes. The unformalized process in regard to market analysis may be the reason that consumers do not take any large part in it. A more formalized process could help the assigned experience planner to be aware of all available information that exists within the Company. This would also decrease the amount of overlap occurring within the company when several people are collecting the same information.

### ***Significant Focus on Pushing Out Features***

Even though Experience Planning collects some of the requirements, without the balance of consumer market data they have a tendency to focus too much on technology, therefore missing the important business value. An interviewee confirms this:

*And when they build feature growth, it is internal innovation, things you push forward.*

Experience Planning have the opportunity to go through with a usability test, testing how future users perceive the software. However, this is not done on a routine basis. Usually, the Usability team asks if Experience Planning needs a test and in other cases planners reach out to Usability, asking for tests. Nevertheless, this is an ad-hoc way of working, and therefore not mandatory.

Both a software project leader and an interviewee from the Usability organization at the Company mention this, especially emphasizing the fact that consumer feedback should be given in the early process of development, since it is very expensive to



Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

change later on when everything is set. This underlines the need for Experience Planning to involve consumer market data, since they provide input at a very early stage. One of our interviewees, responsible for a late phase in the release process, describes this:

*I would say that by then it is pretty set. Things tend to move very quickly and you seldom want to change or rotate the scope. Every change means several weeks of work, and it has direct impact on the launch date. So it is set, you just need to accept that.*

The problem with too much technology focus and internal innovation is aligned with Bjarin's (2013) statement that smartphone manufacturers release more complex features and systems that are not understandable for consumers. Ebert and Brinkkemper (2014) also point out that a too great focus on technology will give a company a collection of features instead of a valuable product. The consumers will probably not perceive the products as valuable and switch to a more consumer-focused product next time they buy a smartphone. The Company should take this into consideration and be more aware of what consumer's value, instead of the engineers working at the Company.

There is also a problem connected with not having a mandatory usability test for software. Rajlich and Bennett (2000) states that it is valuable to be precise in the initial stages of development, since it will be more costly to redesign and redevelop further on in the evolution process. Favaro (2002) highlights that there is a need to have a robust requirement engineering process, meaning you need to include all stakeholders at the beginning, to avoid high cost in later stages of development. This contrasts to the working methods of the Company, where consumer data is not taken in at an early stage but first after the product has been released. Reworks will therefore be more expensive, and because of the financial situation it will not be prioritized and therefore not amended. As a result, the products circulating on the market will probably be seen as less valuable. To take in consumer preferences early in the process should be mandatory so these do not come as a surprise later on.

### ***Experience Planning is Assigned Consumer Responsibility***

An additional problem area arises when a planners tends to think that Experience Planning is the organization, if any, that is responsible for driving the consumer agenda:

*It is more an experience planner who will push the consumer aspect, what do we want to push from our own proposition.*

It is surprising that they should be the organization that pushes the consumer aspect, all the while not even having the consumer aspect as a mandatory part of the market analysis. The consumer perspective is clearly not in focus. They rely on their own ideas and competitive analysis. Nevertheless, they say that they own the user experience:

*We are responsible for all experience aspects of the phone.*

This is problematic, having an organization being responsible for the experience, which do not have a consumer focus.

As mentioned above other organizations should be involved in this process and also be a representative for the consumers.

### ***Discussion***

As stated in this section, there is a strong technology focus with a lot of internal innovation. Consequently, the Company sometimes fails to see if the innovation is valuable to the consumer and may therefore not be able to produce valuable products. There is a need to map these innovations to the market analyses, that today are virtually nonexistent, to see if these features are valuable.

Furthermore, the experience planners do not have any guidelines to follow when collecting market information. Since they are assigned a responsibility to represent consumers, they should collect this kind of information. If they do not have the time or resources available, it is crucial that another organization is assigned consumer responsibility. The consumers need to have some kind of representation at the Company. When an organization is assigned responsibility, evaluation will also be facilitated. However, it is clear that someone needs to be held accountable for

including the consumer perspective as an essential complement to technology innovation.

The market analyses and innovations can of course turn into successful features; however, this should be more accounted to luck than hard labor without integrating consumer market data. It is important for the Company to improve the elicitation activities, since the wrong input equals substandard output, and as previously stated this may be a collection of invaluable features instead of a qualitative and competitive product. In the next section we present a stakeholder that is in some sense a substitute market representative, namely the operators.

#### **4.5.6 Operators Are Seen as the Most Important Stakeholder**

Operators were frequently identified as important and influential stakeholders by the interviewees. Moreover, one main finding in the interviews at the Company is that operators have a loud voice in comparison with other stakeholders. This could be the result of several factors, and it is a problem area for the Company.

##### ***Several Internal Organizations Represent the Operators***

Operators are represented by three different organizations at the company. One organization has a long-term strategic vision, linking operator technology with company technology. The Operator Requirement Manager works with requirement specifications in Focal Point, aiming at ensuring functionality in the networks and managing operator specific features. Finally, Technical Sales are responsible for the operational contact and for testing and releases. Due to the many ways of contact, several groups in the organizations are responsible for being the operator representative in-house, as mentioned above operators even have a manager responsible for translating their requirements.

The organizations representing the operators in-house have specific methods of getting top priority, which is only available to them. For example Technical Sales is the only organization able to set a tag in the quality improvement system that will immediately hinder the software launch, if the issue is not corrected:

*We can put a showstopper tag on issues, if we find it extremely important, and we have done this on a few occasions. When we use this tag, we are usually heard and*

*prioritized. I think this is mainly because we are interfacing the operators, rather than us facing the market.*

### **Operators Are a Substitute for the Consumers**

Operators receive a lot of complaints regarding defects, through their Customer Service channels. Therefore it is another important stakeholder to gather requirements from. But according to one of the interviewees they do not always have the same agenda as the consumer and therefore it is necessary to complement their view with the consumer perspective. One interviewee highlights the problem from another point of view:

*But of course, we also get a lot of input from the market. The company has a lot of cooperation with operators, they tell us for example what features they want to see.*

Hence, operators are the only market representative, which implies that there is no interest in consumers and addressing their requirements.

In van Weerd et al's (2006) framework, partner companies should come with different kinds of input, both market trend identification and partner requests, just as the operators are at the Company. Since it is a market-driven context, it can also be hard to collect the voices of several millions of consumers. This is a possible reason for the prominent role of the operator. They are easier to understand and to contact than millions of consumers. Listening to stakeholders that are more easily reached and substituting vague and inaccessible requirements with those that are easily interpreted and accessible is a common problem according to Pacheo and Garcia (2012). In this case, the consumer voice is substituted by the voice of the operators.

### **Gatekeepers to the Market and Direct Revenue Source**

Partner companies are one of the important stakeholders in software product management according to van Weerd et al (2006), but the involvement should be combined with the interaction of other stakeholders, this stands in contrast to how the Company currently works. There are several factors causing this, one interviewee blames tradition:

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

*Historically, operators have had very loud voice. It's one of our main problems; we have focused on what the operators want and not what the consumer wants. We are trying to change that.*

Another interviewee claims that it is the operators who pay their salaries:

*The operators we listen to a lot, I'm not saying that it is wrong; it is the operators who pay our salaries. Sometimes we forget the consumers.*

It is also very important to take into consideration the requirements of the operator, in order for the phones to work in their network configurations, as this will also affect the end-users perception of how the smartphone is used.

The comment that the operators pay their salaries is also consistent with the operators in fact being the main revenue source for the company. Another employee takes another perspective on this, and claims that if you win the consumers the operators will follow the consumer market. The fact that most of the phones are sold through operators is important to consider, and it should in fact be an important perspective to consider.

Compared to van Weerd et al.'s (2006) framework, they can also be seen as the customer. Therefore they are two stakeholders in one, which may be a possible reason why they are frequently mentioned.

### **Discussion**

It is important to listen to operators. However, operators should be seen as a partner, instead of a customer. As partners they should work towards delivering value to the consumers together.

In some ways the Company substitutes the market input with the operators' opinions, and several employees believe this needs to change. The operators do not have the same agenda as the consumers at all times. However, it can be understood that they are listened to so much since the company seeks input from someone, and the operators are much more accessible compared to consumers. They are gathered and deliver the requirements in a direct way.

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

In-house organizations representing operators are also numerous and strong compared to the ones representing the consumer. These organizations also have some privileges in prioritization. In order to listen more actively to other stakeholders this advantage should also be available for organizations connected to consumers. Since the operator have their own requirement manager, one person states that it would be a possibility that the consumers had one as well.

## 4.6 Summary of the Case Study Findings

The main problems at the Company, the key issues and the corresponding problems identified in previous research are summarized in table 5 below.

Problem area	Corresponding previous research	Key takeaways
<b>Lack of a common vision</b>	No clear vision, root cause to wrong content, rework and scope creep (Ebert, 2009)	<ul style="list-style-type: none"> <li>• There is no uniform vision properly communicated through the company</li> <li>• A common vision - an instrument steering them in the same direction</li> </ul>
<b>Non-existing software product management role</b>	No interface between business and technology functions (Ebert, 2007; Fricker, 2012), no alignment of the three value perspective product manager (Eber & Brinkkemper, 2014)	<ul style="list-style-type: none"> <li>• No authority over the lifecycle of the software, governance become difficult</li> <li>• Working with a software product manager is complicated due to the embedded product, a need to unite the aspects of the embedded product: hardware, software and business</li> <li>• Difficult to align value creation</li> </ul>
<b>Unpredictable and non-transparent requirement prioritization</b>	Value neutral requirement engineering setting (Boehm, 2005), difficulties in estimating cost and value (Karlsson et al. 2007)	<ul style="list-style-type: none"> <li>• The unclear prioritization raises suspicion within other organizations</li> <li>• Prioritizations need to be motivated more clearly, underlining the need of evaluating business case and releases</li> <li>• Estimating benefits are difficult due to the market-driven context and the embedded product</li> <li>• The lack of vision likely contributes to the perception of obscurity</li> </ul>
<b>Consumer awareness, but no consumer focus</b>	Silowork (Ebert, 2014)	<ul style="list-style-type: none"> <li>• Difficult to produce software that generates value, since there is no consumer focus of development</li> <li>• Gathered consumer information is not properly used</li> <li>• Estimating benefits is difficult due to the market-driven context</li> </ul>

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

		<ul style="list-style-type: none"> <li>• Consumer requirements are difficult to understand, and therefore their needs are not sufficiently understood</li> </ul>
<b>Imbalance between idea generation and market analysis in elicitation activities</b>	Imbalance in market pull and technology push, requirements are invented by engineers (Regnell and Brinkkemper, 2005; Karlsson et al. 2007) Too much focus on technology (Ebert, 2007)	<ul style="list-style-type: none"> <li>• Innovations are not properly mapped against the consumers' needs</li> <li>• No guidelines for experience planners in market analyses - performed at an ad hoc basis</li> <li>• It is important that elicitation activities are improved, as the wrong input will result in an inferior product</li> </ul>
<b>Operators are seen as the most important stakeholder</b>	Substituting hard to get sources of requirements with easier sources (Pacheco and Garcia, 2012)	<ul style="list-style-type: none"> <li>• Operators should be used as a complement to the consumer market input, not as a substitute</li> <li>• Consumer voice needs to become stronger and more accessible</li> <li>• Operators are important partners, and the Company should focus on establishing the partnership</li> </ul>

**Table 5. Summary of problem areas, related to previous research.**

Several of the problem areas at the Company have been seen in previous research. We have observed a significant technology focus in the Company, and an imbalance in market analyses. This problem is pointed out in both software product management (Ebert, 2014) and in market-driven requirement engineering (Regnell and Brinkkemper, 2005; Karlsson et al. 2007; El Emam and Madhavji, 1995).

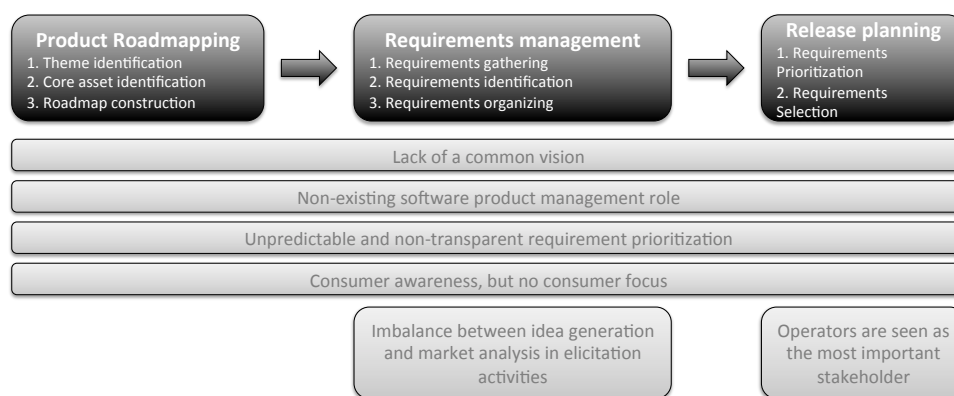
Additionally, we have seen problems related to the communication between organizations and that the Company has problems establishing a foundation for communication between development and organizations working with consumers. Karlsson et al. (2007) emphasize the problem in a requirements engineering perspective. Ebert (2014) acknowledges that this is often the case if there is an insufficient software product management mindset. Furthermore, there is a difficulty in reorganizing the way of working with requirements engineering at companies (see Karlsson et al, 2007). This is also the case at the Company.

Interestingly, we mostly found problems resulting from organizational and social aspects (see Karlsson et al 2007; Hall et al. 2002). This indicates that management is required in the area. This management perspective could be added through software product management, which could enable alignment.

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

Finally, we have linked the identified problem to the three main activities in software product management, in Figure 17 below. There are several areas that span all three stages, meaning that the effects are to some extent present in all aspects of the processes. Also, two of the problem areas are more specifically linked to activities of requirements management and release planning.



**Figure 17. Problem areas connected to the three main activities in software product management**

The likely effect of the problem areas that we have identified is a product, which is not as valuable as it could be. It is apparent that addressing the problem areas would have significant effects on the over-all performance of the process as a whole, as well as in particular sub-processes.



## 5 Conclusions and Final Remarks

---

*In this concluding chapter we answer our research questions that were stated in the first chapter. Moreover, we give recommendations to the Company, based on the findings in the analysis. We discuss our results from an academic point of view and make suggestions for future research areas.*

---

### 5.1 Introduction

The purpose of this thesis has been to examine which problem areas a smartphone manufacturer face from a software product management perspective and thereby increase the understanding of Software product management in the smartphone context. We will in this section describe and discuss the answers to our research questions.

#### ***How do the described processes at the Company relate to the processes described in Software Product Management?***

Currently, there is no explicit way of working with software product management. However, there are certain elements that are recognizable. The software scope planner has some of the responsibilities the software product manager described in previous research has, however not all, which is shown in figure 16. Furthermore, the three main processes in software product management examined in this thesis: road-mapping, release planning and requirement engineering are also to some extent present and they are performed in the same order as in theory. However, there are several problem areas in the processes and they are essentially not performed in accordance to a theoretical framework. In figure 17 the problem areas are mapped against the main activities of software product management.

The problem areas span across the entire process, this suggests that having a team or a role to govern the product lifecycle in order to generate the biggest possible value to the business would be helpful. If the products were to be produced in a more valuable way from a software product management perspective, the software would likely be more than a mere feature collection and the product value to consumers would increase.

***Which are the problem areas that the Company encounters from a software product management and a market-driven requirements engineering perspective?***

We have found several problem areas in the Company processes, reflecting previous research both in software product management and market-driven requirements engineering, this are concluded table 5. These are namely *Lack of a common vision, the non-existing software product management role, unpredictable and non-transparent requirement prioritization, consumer awareness - but no consumer focus, imbalance between idea generation and market analysis in elicitation activities* and *operators are seen as the most important stakeholder*. Many of the problem areas found in our study resemble problems described in previous research, implying that companies in these industries have similar problem areas to work with. However, the operator being seen as the most important stakeholder is very specific to the smartphone industry.

## 5.2 Recommendations for the Company

The recommendations to the Company are based on the problem areas we have found, and can be helpful to overcome the main problems. The recommendations are foremost based on ideas from the theoretical framework, but also complemented with our own thoughts, pointing out how the problem areas in software product management and especially problems in requirements engineering can be resolved. Hopefully these recommendations can act as guidelines to the Company. The proposed remedies are summarized in table 6.

Problem areas	Proposed remedies
<b>Lack of a common vision</b>	<ul style="list-style-type: none"> <li>• Entitle a software product manager to create a software vision, aligned with overall goals and market demands</li> <li>• Involve all necessary stakeholders in making the vision to secure that all important perspectives are fully addressed</li> <li>• Communicate the software vision to all stakeholders included in the development process</li> </ul>
<b>Non-existing software product management role</b>	<ul style="list-style-type: none"> <li>• Appoint a role with authority over the entire lifecycle</li> <li>• Unite all aspects of the embedded product: hardware, software <i>and</i> business</li> </ul>
<b>Unpredictable and non-</b>	<ul style="list-style-type: none"> <li>• Increase knowledge and information transfer in prioritization processes to lower the suspicion against the prioritization process</li> </ul>

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

<b>transparent requirement prioritization</b>	<ul style="list-style-type: none"> <li>• Reduce ad-hoc decisions by formalizing the processes making it evident who is responsible for prioritizing and making decisions</li> <li>• Ensure that development teams are familiar with all perspectives or give all prioritization power to the product scope planner who should make sure he has knowledge of all the stakeholders value perspectives</li> <li>• Put more effort into evaluating how updates are received in the market and direct resources so that this can be done</li> <li>• Create business cases that can be used to evaluate if an update should be released</li> </ul>
<b>Consumer awareness, but no consumer focus</b>	<ul style="list-style-type: none"> <li>• Be alert and integrate the consumer perspective in development processes</li> <li>• Focus on spreading the word that consumers should always be the top priority, and since operators have their own requirements translator the same should be possible for consumers</li> <li>• Encourage consumer curiosity within organizations responsible for prioritizing requirements</li> </ul>
<b>Imbalance between idea generation and market analysis in elicitation activities</b>	<ul style="list-style-type: none"> <li>• Map internal innovations to consumer preferences and needs</li> <li>• Create awareness of what consumers value and emphasize the importance of considering this in market analyses</li> <li>• Formalize contents of market analyses, so all relevant information is gathered</li> <li>• Including consumer preferences early in processes should be mandatory reducing the need for costly reworks</li> <li>• Have more organizations than Experience Planning representing the consumer</li> </ul>
<b>Operators are seen as the most important stakeholder</b>	<ul style="list-style-type: none"> <li>• Use operators as a complement to the consumer market input, not as a substitute</li> <li>• Increase the consumer voice, by giving consumer organizations more authority and enabling communication with development</li> <li>• Establish partnerships with operators and deliver value together with the operators, as opposed to seeing them as customers</li> </ul>

**Table 6. Summarizing table over possible remedies to the Company, divided upon the found problem areas, and on long-term and short-term.**

### 5.3 Discussion of Results

The findings in this study can be used as guidelines for the Company to identify the problem areas that hinder them from creating valuable software. Possibly, companies that operate in similar settings could also use these guidelines.

Although we are satisfied with the results, there may have been problem areas that we did not observe. The deliberate open research approach and the explorative nature of the study did however limit this risk. Due to the said approach, we were

able to observe a broad range of problems at the Company. We chose not to present all problem areas seen in the study, since the limited time frame required us to focus our efforts where we saw the most potential benefit, both practically and theoretically. Most of the areas that we excluded were not as clearly connected to others as the ones presented here, and were in some cases likely to be very company-specific. However, these areas could have been valuable to the Company or to companies in similar settings. Future studies may be able to fill this gap.

Furthermore, even though we are confident that we have provided a fairly in-depth analysis and description of the identified problem areas, it would be beneficial to evaluate and examine the problems even deeper. The problem areas could be studied further at the Company or at other companies experiencing similar issues. Such analysis could provide a more comprehensive analysis of root causes and effects, and how to best mitigate them.

The primary aim of this study was never to produce results that could be widely generalized, and as a result, we do not believe that they are strictly generalizable. However, they could apply to companies in similar settings. As stated in the Methodology chapter, the generalizability of the study is first and foremost a matter of future research. Consequently, we have compiled our suggestions for future research, which are presented below.

### **5.3.1 Theoretical Contribution**

We believe that our study contributes to previous research in both software product management and market-driven requirements engineering by providing a new perspective, and testing the areas in a new context. Through our case study findings we are able to validate some of the problems stated in previous research, add further nuances to previously identified problems, as well as uncovering one specific problem area in the smartphone manufacturer context.

Despite the particular context, we consider it very likely that conducting the same study at another company with the same prerequisites would generate very similar results. Therefore, our study contributes both to the Company, as well as other companies.

### 5.3.2 Suggestions for Future Research

One significant problem that we encountered when we conducted the interviews was that we could not secure that the interviewee was telling us how they would like to work or how they actually worked. We tried to be as precise in our questioning as possible and asked for examples of how things could have been done. An improvement of the study could be to follow a specific release throughout the process, with interviews, observations in meetings and looking at documents and thereafter frame a more viable answer.

We have conducted a single case study; to further analyze this phenomenon a multiple case study would validate and make the results generalizable. It could either be carried out with several companies, or conducting interviews more with staff in the same positions from the same company, mostly spanning medium- and large-sized companies. It would also be interesting to do a study including not only smartphone manufacturers, but also other industries manufacturing products with embedded software. This would result in a more generalizable framework for market-driven software product management within embedded products.

When constructing the interview guide it would have been interesting to ask the interviewees further on what they saw as problems and how they believed these could be improved or resolved. Also, we suggest that in-depth studies of the specific identified problem areas are conducted, which would enable a more thorough understanding of root causes and remedies. Furthermore, evaluating the actual impact of the problem areas would also constitute an interesting study, which would also enable an understanding of which problem area would be the most important to address.

As we can see there is a two-fold explanation to why the Company's work methods differ from the one described in the theoretical framework. There is as emphasized before a difficulty in the embedded product context where the software is just one aspects of the value proposition. Therefore, a software product manager completely responsible for the whole product is not possible. There needs to be close collaboration between hardware and software. Furthermore, they have not realized that there is a need for change yet. Thus, it would also be interesting to further research what product management should be for embedded products.

## 5.4 Concluding Remark

In conclusion, this thesis contributes to an immature research area by adding new perspectives, emphasizing a market-driven context where software is embedded in a hardware product. Moreover, our findings enhance the understanding of requirements engineering and software product management by validating results from previous industrial surveys.

Researchers could use the results when identifying new research areas. Furthermore, the findings are important guidelines for practitioners, as they can help identifying obstacles in creating valuable software in the particular context. We make a significant contribution to the company by identifying their improvement potential, as well as suggesting how to overcome their issues. If adopted, it is very likely that the approaches suggested in this thesis will enable them to produce more profitable products, and potentially reverse their recent downward spiral.

Finally, software product management clearly needs to be complemented with the perspectives presented in this report. As software becomes widely spread throughout our everyday lives, the area of software product management for embedded products in the market-driven context becomes increasingly important, and a potentially powerful tool for generating profitable products in a competitive environment.

## 6 Bibliography

- Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. N. (2014). A systematic literature review of software requirements prioritization research. *Information and Software Technology*, 56(6), 568-585. doi: 10.1016/j.infsof.2014.02.001
- Alwis, D., Hlupic, V., & Fitzgerald, G. (2003). Intellectual capital factors that impact of value creation. *Proceedings of the 25th International Conference on Information Technology Interfaces, 2003 (ITI 2003)*, 411.
- Anderson, J. C., & Narus, J. A. (1998). Business marketing: understand what customers value. *Harvard business review*, 76, 53-67.
- Aurum, A., & Wohlin, C. (2005). *Aligning requirements with business objectives: a framework for requirements engineering decisions*. Paper presented at the Workshop on Requirements Engineering Decision Support, REDECS'05, Paris, France.
- Aurum, A., & Wohlin, C. (2007). *A value-based approach in requirements engineering: Explaining some of the fundamental concepts*. Paper presented at the Lecture Notes in Computer Science. Conference Paper retrieved from <http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-38149055748&site=eds-live&scope=site>
- Barney, S., Aurum, A., & Wohlin, C. (2008). A product management challenge: Creating software product value through requirements selection. *Journal of Systems Architecture*, 54(6), 576-593. doi: 10.1016/j.sysarc.2007.12.004
- Bekkers, W., van de Weerd, I., Spruit, M., & Brinkkemper, S. (2010). A framework for process improvement in software product management *Systems, software and services process improvement* (pp. 1-12): Springer.
- Berander, P., Khan, K. A., & Lehtola, L. (2006). *Towards a research framework on requirements prioritization*. Paper presented at the Proceedings of Sixth Conference on Software Engineering and Practice in Sweden (SERPS'06).
- Black, S. (2006). The role of ripple effect in software evolution. *Software evolution and feedback*, 249.
- Boehm, B. W. (2006). Value-based software engineering: Overview and agenda (pp. 3-14): Springer Berlin Heidelberg.
- Bryman, A., & Bell, E. (2007). *Business research methods / Alan Bryman, Emma Bell*: Oxford : Oxford University Press, cop. 2007  
2. ed.
- Carlshamre, P. (2002). Release Planning in Market-Driven Software Product Development: Provoking an Understanding. *Requirements Engineering*, 7(3), 139-151.

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

- Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., & Natt och Dag, J. (2001, 01//). *An industrial survey of requirements interdependencies in software product release planning*. Paper presented at the Proceedings Fifth IEEE International Symposium on Requirements Engineering.
- Dedrick, J., Kraemer, K. L., & Linden, G. (2011). The distribution of value in the mobile phone supply chain. *Telecommunications Policy*, 35, 505-521. doi: 10.1016/j.telpol.2011.04.006
- Ebert, C. (2007). The impacts of software product management. *Journal of Systems & Software*, 80(6), 850-861. doi: 10.1016/j.jss.2006.09.017
- Ebert, C. (2009). Software product management. *CrossTalk*, 22(1), 15-19.
- Ebert, C., & Brinkkemper, S. (2014). Software product management - An industry evaluation. *Journal of Systems and Software*. doi: 10.1016/j.jss.2013.12.042
- El Emam, K., & Madhavji, N. H. (1995). *A field study of requirements engineering practices in information systems development*. Paper presented at the Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on.
- Favaro, J. (2002). Managing Requirements for Business Value. *IEEE Software*, 19(2), 15-17.
- Fricker, S. A. (2012). Software Product Management Software for People (9783642313707) (pp. 53-81).  
[http://www.bth.se/fou/forskinfor/nsf/all/7f6d9ec8bf616025c1257ad0004b5546/\\$file/Fricker\\_SPM\\_published.pdf](http://www.bth.se/fou/forskinfor/nsf/all/7f6d9ec8bf616025c1257ad0004b5546/$file/Fricker_SPM_published.pdf) Retrieved from  
<http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=94108113&site=eds-live&scope=site>.
- Glinz, M., & Wieringa, R. J. (2007). Stakeholders in requirements engineering. *IEEE Software*, 24(2), 18-20. doi: 10.1109/MS.2007.42
- Grabham, D. (2012, August, 28). Smartphone software updates explained Retrieved 12 february, 2014, from <http://www.techradar.com/news/phone-and-communications/mobile-phones/smartphone-software-updates-explained-1093733>
- Grunbacher, P., & Briggs, R. O. (2001). Surfacing tacit knowledge in requirements negotiation: experiences using EasyWinWin. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, 8.
- Hacklin, F., Battistini, B., & Von Krogh, G. (2013). Strategic choices in converging industries. *MIT Sloan Management Review*, 55(1), 65-73.
- Hall, T., Beecham, S., & Rainer, A. (2002). Requirements problems in twelve software companies: an empirical analysis. *IEE Proceedings-Software*, 149(5), 153-160.
- Haron, A., & Sahibuddin, S. (2010). The strength and weakness of Requirement Engineering (RE) process. *2010 2nd International Conference on Computer Technology & Development (ICCTD)*, 56.



Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

- Henneberg, S., Pardo, C., Mouzas, S., & Naudé, P. (2005). *Value dimensions and strategies in dyadic 'key relationship programmes'*. Paper presented at the Proceedings on the 21st IMP Conference.
- ISPMA. (2013). *SPMBOK: Software Product Management Body of Knowledge*. Retrieved from <http://ispma.org/spmbok/>
- Jacobsen, D. I. (2002). *Vad, hur och varför : om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen / Dag Ingvar Jacobsen ; översättning: Gunnar Sandin ; bearbetning av den svenska upplagan: Caroline Hellström: Lund : Studentlitteratur, 2002 (Lund : Studentlitteratur).*
- Jain, A., & Boehm, B. (2005). Developing a theory of value-based software engineering. *ACM SIGSOFT Software Engineering Notes*, 30(4), 1-5.
- Karlsson, L., Regnell, B., Natt och Dag, J., Dahlstedt, Å. G., & Persson, A. (2007). Requirements engineering challenges in market-driven software development - An interview study with practitioners. *Information and Software Technology*, 49(6), 588-604. doi: 10.1016/j.infsof.2007.02.008
- Keller, E. (1995). The value of software. *Manufacturing Systems, ABI/INFORM Global*, 13(1), 16.
- Kenney, M., & Pon, B. (2011). Structuring the Smartphone Industry: Is the Mobile Internet OS Platform the Key? *Journal of Industry, Competition and Trade*, 11(3), 239-261. doi: 10.1007/s10842-011-0105-6
- Kittlaus, H.-B., & Clough, P. N. (2009). *Software Product Management and Pricing [Elektronisk resurs] : Key Success Factors for Software Organizations / by Hans-Bernd Kittlaus, Peter N. Clough: Berlin, Heidelberg : Springer Berlin Heidelberg, 2009.*
- Konig, S. J. (2009). *Finance as a stakeholder in product management*. Paper presented at the Software Product Management (IWSPM), 2009 Third International Workshop on.
- Lawrence, B., Wiegers, K., & Ebert, C. (2001). The Top Risks of Requirements Engineering. *IEEE Software*, 18(6), 62.
- Maglyas, A., Nikula, U., & Smolander, K. (2011, 2011 / 01 / 01 /). *What do we know about software product management? A systematic mapping study.*
- Maglyas, A., Nikula, U., & Smolander, K. (2013). What are the roles of software product managers? An empirical investigation. *Journal of Systems and Software*, 86(12), 3071-3090. doi: 10.1016/j.jss.2013.07.045
- Maurice, S., Ruhe, G., Saliu, O., & Ngo-The, A. (2006). Decision support for value based software release planning *Value-Based Software Engineering* (pp. 247-261). Berlin Heidelberg: Spring.
- Murphy, W. H., & Gorchels, L. (1996). How to improve product management effectiveness. *Industrial Marketing Management*, 25(1), 47-58.
- Pacheco, C., & Garcia, I. (2012). A systematic literature review of stakeholder identification methods in requirements elicitation. *Journal of Systems and Software*, 85(9), 2171-2181.

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

- Pressman, R. S. (1997). *Software Engineering: A practitioner's approach. McGraw Hill International Edition*, 466-472.
- Rajlich, V. T., & Bennett, K. H. (2000). A staged model for the software life cycle. *Computer*, 33(7), 66-71.
- Regnell, B., & Brinkkemper, S. (2005). Market-driven requirements engineering for software products *Engineering and Managing Software Requirements* (pp. 287-308).  
: Springer Berlin Heidelberg.
- Regnell, B., Höst, M., Natt och Dag, J., Beremark, P., & Hjelm, T. (2000). An industrial case study on distributed prioritization in market-driven requirements engineering for packaged software. *Requirements Engineering*, 6(1), 51-62
- .
- Ruhe, G., Eberlein, A., & Pfahl, D. (2003). Trade-off analysis for requirements selection *International Journal of Software Engineering and Knowledge Engineering*, 13(4), 345-366.
- Shaw, M. (2002). *Everyday dependability for everyday needs*. Paper presented at the Supplemental Proc of 13th International Symposium on Software Reliability Engineering, Maryland, US.
- Smith-Strickland, K. (2013). The 10 worst things you're doing with your smartphone. Retrieved 23 februari, 2014, from <http://www.popularmechanics.com/technology/how-to/tips/the-10-worst-things-youre-doing-with-your-smartphone-15216505>
- Van De Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., & Bijlsma, L. (2006, 2006 / 01 / 01 /). *Towards a reference framework for software product management*.
- Weinstein, A. (1999). *Designing and delivering superior customer value: concepts, cases, and applications*: CRC press.
- Wohlin, C., & Aurum, A. (2005). Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study. In S. Biffl, A. Aurum, B. Boehm, H. Erdogan & P. Grünbacher (Eds.), *Value-based Software Engineering*: Springer Verlag.
- Yin, R. K. (2007). *Fallstudier : design och genomförande / Robert K. Yin ; i översättning av Björn Nilsson ; [fackgranskning: Jan Teorell]*: Malmö : Liber, cop. 2007 (Slovenien) 1. uppl.
- Yin, R. K. (2013). *Kvalitativ forskning från start till mål / Robert K. Yin ; översättning: Joachim Retzlaff ; svensk bearbetning: Björn Östbring*: Lund : Studentlitteratur, 2013 (Spanien) 1. uppl.

## 7 Appendix I - Interview Guide

### Introductory questions

- What is your role?
- What does your organization do?
- What does your organization do regarding software releases?
- What is your role regarding software releases?
- Do you have a formal process for software releases?
  - If "yes":
    - Is this process followed?
    - Is there a process owner?
      - If "yes":
        - Who owns the process?
  - If "no":
    - Is there an informal process?
    - Is the informal process always followed?
    -

### Requirement elicitation/gathering

- How do you gather new requirements?
- Which roles or organizations are involved and in which way?
- Is there a formal way of gathering requirements?

### Requirement interpretation, structuring, verification and validation

- How do you validate requirements?
- Who interpret the requirements?
- How do you document this?

### Requirement prioritization and selection

- Who is responsible for prioritizing requirements?
- Who is responsible for selecting the requirements to include in the release?
- Which tools do you use for this?
- In which forum is the prioritization conducted?
- Do you use formal methods, such as a cost-benefit analysis?
- Do you use informal methods?

Problem Areas in Creating Valuable Smartphone Software  
- A Software Product Management Case Study

---

- Could you identify the stakeholders involved and rank them in terms of their impact on selection and prioritization of requirements?
- What is taken into account when selecting requirements?
- Are there times when you do not see eye to eye with other stakeholders?
- What is the cause of the disagreement?
- How do you handle this?

**Concluding questions**

- How and when do you plan software releases?
- What is the purpose of updating?