Heli Päätalo

STAKEHOLDER INTERACTIONS IN CROSS-FUNCTIONAL PRODUCTIZATION

THE CASE OF MOBILE SOFTWARE DEVELOPMENT
HEL I PÄÄTA LO

STAKEHOLDER INTERACTIONS IN CROSS-FUNCTIONAL PRODUCTIZATION
The case of mobile software development

Academic dissertation to be presented with the assent of the Doctoral Training Committee of Technology and Natural Sciences of the University of Oulu for public defence in the OP auditorium (L10), Linnanmaa, on 26 September 2014, at 12 noon

UNIVERSITY OF OUL U, OULU 2014
This study examines stakeholder interactions in the context of cross-functional software productization. This research aims to increase theoretical and empirical understanding of interactions in software productization.

The theoretical framework consisted of the stakeholders of software productization, their key functions and the elements of interactions such as communication, collaboration, and integration. The empirical data was gathered through interviews, company documentation and participant observation in the case organization. The research strategy is a qualitative case study including some elements of action research.

As the main finding of this study, a model was developed of organization internal stakeholder interactions in the context of software productization. The model consists of identified stakeholders, the elements of interaction and their intensity. In order to create successful software products, it is increasingly important to acknowledge, consider and meet stakeholders’ views, needs and expectations.

According to the findings, the stakeholders are divided into two groups. The first consists of stakeholders who are responsible for decisions related to the company product portfolio. The second group of stakeholders is focused on software product development and implementation. Within the groups, interaction is intensive, including communication, collaboration and integration. According to the data, interactions between groups are one-way only and often carried out via documentation. The stakeholder group responsible for product-related decisions bases its interaction on communication which is perceived by the other group as one-way and unreciprocated. This does not promote dialogue between stakeholders, which is essential for achieving common understanding.

This study contributes and adds to the limited empirical research on software productization. The new knowledge of stakeholder interactions in cross-functional productization supports project managers in the development of approaches in the understanding of stakeholders’ needs and expectations. From a managerial viewpoint, this study discusses a number of means which can be used in software productization.

Keywords: collaboration, communication, integration, interaction, productization, stakeholder
Päättalo, Heli, Sidosryhmien vuorovaikutus ohjelmistoratkaisujen tuotteistamisessa. Mobiiliratkaisujen tapaus


Tutkimus paljastaa, että sidosryhmien vuorovaikutus on monimutkaisempaa kuin kirjallisuudessa on kuvattu. Se täydentää vielä varsin harvalukuisia tuotteistamiseen liittyviä tutkimuksia. Käytännön suosituksesta tutkimus esittää joukon keinoja, joita ohjelmistojen tuotteistamisessa voidaan hyödyntää.

Asiakohdin: integrointi, kommunikointi, sidosryhmät, tuotteistaminen, vuorovaikutus, yhteistyö
Dedicated to my little sister, Tuija Päätalo
Acknowledgements

I would like to present my thanks to Professor Casper Lassenius and Professor Jürgen Münch, the reviewers of this thesis, for taking the time to analyze and give critical comments and point out areas of improvement, providing me with a chance to develop myself. My warmest thanks to my supervisors, Professor Veikko Seppänen and Dr. Juhani Warsta, who guided me from the chaos of beginning to the finish line in a firm but always friendly way. I will miss our vivid monthly discussions.

I also want to thank Nokia and my former colleagues there for their support of my studies during these past years, and the interviewees for their time and openness.

I would like to thank my dear friend, Dr. Riitta Laulajainen, who has shared and acted as a peer group during my journey, and with whom I have been able to share the moments of accomplishment and downsides of writing this study. You have always found some words of encouragement, and as you can see, it has helped a great deal.

Special thanks to my Rottweiler, Kaapo, who has contributed to this thesis by supervising, making sure that I am at my desk and ensuring that I have taken a few breaks by interrupting me whenever he has found it necessary.

Finally, I want to thank my mother Leena, father Matti and my spouse Jyyne. Thank you for standing by me and for your loving support.

Kempele, 12.1.2014

Heli Päätalo
# Table of contents

**Abstract**

Tiivistelmä

Acknowledgements 9

Table of contents 11

1 Introduction 15

1.1 Research objectives and motivation ........................................................ 16

1.2 Research problem and questions .............................................................. 19

1.3 Research approach .................................................................................. 19

1.4 Prior literature review ............................................................................. 21

1.4.1 Stakeholder research ..................................................................... 21

1.4.2 Interactions ................................................................................... 28

1.4.3 Productization ............................................................................... 32

1.4.4 Mobile software development ...................................................... 35

1.4.5 Conclusions of the literature review .................................................................. 38

2 Mobile software development 39

2.1 Mobile market ......................................................................................... 39

2.2 Characteristics of mobile software ........................................................ 41

2.2.1 Software product line approach .................................................... 43

2.2.2 Distributed software development ................................................ 44

2.2.3 Open software ecosystems ........................................................... 46

2.3 Mobile software product development .................................................... 46

2.4 Stakeholders of software product development ...................................... 49

2.4.1 Marketing ..................................................................................... 51

2.4.2 Corporate management ................................................................. 53

2.4.3 Software product management ....................................................... 55

2.4.4 Platform development .................................................................... 62

2.4.5 Product development .................................................................... 64

3 Productization 69

3.1 The concept of productization ................................................................. 69

3.2 Conceptual model of productization ...................................................... 72

3.3 Productization through modularity ...................................................... 74

4 Interaction 75

4.1 Introduction ............................................................................................. 75

4.2 Stakeholder interactions .......................................................................... 75

4.3 Elements of interaction ............................................................................. 79
4.3.1 Communication ................................................. 80
4.3.2 Collaboration ..................................................... 82
4.3.3 Integration .......................................................... 85

5 Reference framework ................................................. 87
5.1 The a priori model ..................................................... 87
  5.1.1 Mobile market needs and opportunities ................. 90
  5.1.2 Stakeholders ....................................................... 90
  5.1.3 Cross-functional software productization ............... 94
  5.1.4 Interactions ........................................................ 94
5.2 Summary .................................................................. 97

6 Research design .......................................................... 99
6.1 Ontology, epistemology and methodology ................. 99
6.2 Qualitative case research ....................................... 101
6.3 Research process and empirical data analysis ............. 112
  6.3.1 Research process and empirical data analysis ........ 112
  6.3.2 Analysis of the empirical data .............................. 114
6.4 Limitations of the research ....................................... 116
6.5 Summary ................................................................. 116

7 Analysis of stakeholder interactions in mobile software
  productization .......................................................... 117
7.1 The case organization ............................................. 117
7.2 Cross-functional mobile software productization ........ 120
7.3 Stakeholders of software productization .................... 123
7.4 Interactions in cross-functional software productization 136
  7.4.1 Communication ............................................... 136
  7.4.2 Collaboration .................................................... 145
7.5 Summary ................................................................. 150

8 Discussion ................................................................. 151
8.1 Stakeholder interactions in cross-functional software
  productization ......................................................... 151
  8.1.1 Interactions ....................................................... 155
  8.1.2 Stakeholders of software productization ............... 160
8.2 Theoretical contributions ........................................ 163
8.3 Managerial implications ......................................... 168
8.4 Limitations of the research ...................................... 173
8.5 Suggestions for future research ............................... 174
9 Conclusions

9.1 Answers to the research questions ........................................................ 175
9.2 Concluding words ................................................................................. 179

References 181
Appendices 207
1 Introduction

Companies must be able to manage both their internal and external stakeholders and stakeholder groups in order to be able to deliver successful products. Numerous stakeholders are involved and impact on a company’s new product development. They contribute their needs, expectations, knowledge, and support through interacting with product development, but also with each other. Considering and balancing all of the influencing stakeholders’ demands is a difficult task for a product development project manager. If the expectations of stakeholders are ignored, there is a high risk that the project will not be successful, even if the outcome is delivered within budget, scope, and time (Bourne & Walker 2005). Not much is known about managing multiple stakeholder demands and issues during new product development, but positive impacts on development performance emerge when stakeholders’ issues are addressed (Driessen & Hillebrand 2010).

The competition for mobile device industry market share and the demand for faster time-to-market has increased drastically. There is an identifiable trend for shortened life cycles in consumer electronics (Minderhout & Fraser 2005) because of technological advances, especially in the mobile phone market (Reiner et al. 2009). The need to accelerate the development time has never been greater and the time seems to constantly shorten (Smith & Reinertsen 1992). Mobile device companies aim to achieve the goals of the company and satisfy their shareholders by increasing their market share through the means of reaching new users and market segments and providing desirable products which stand out from their competitors. It is characteristic of the mobile branch of the industry that the market is very demanding and waits impatiently for functional innovations and new technologies. Thanks to the fast life cycle of mobile devices, it is essential that companies have the capability to manage and rapidly respond to changing customer needs and wishes. Companies are investing a great deal of resources into seeking means for faster development methods and decision-making in their R&D development.

The purpose of this research is to examine stakeholder interactions in cross-functional software productization in the context of mobile software development. Nokia will be referred to as a case company. The dissertation is set out as follows:

Chapter 1 introduces the research objectives, the research questions, the research approach and the literature review related to this topic. The scope of the
research themes that links stakeholders, interactions and productization is identified.

Chapter 2 introduces the contextual view on mobile software development and its characteristics, and serves as a foundation for the following core chapters.

Chapter 3 explains the theoretical aspects of productization which are central components of this research. This chapter provides the overview to the circumstances of the research problem.

Chapter 4 includes a theoretical perspective of interaction and related elements.

Chapter 5 introduces the reference framework. The established a priori model is based on Chapters 2, 3 and 4, to guide the data collection and analysis.

Chapter 6 explains the research design, and describes the philosophical basis of the research approach, the research strategy, and the empirical techniques.

Chapter 7 presents the empirical data and the analysis of stakeholder interactions in mobile software productization.

Chapter 8 discusses the results of this study.

Chapter 9 summarizes the answers to the research questions, and the managerial implications. Directions for future studies of the topic are speculated upon.

1.1 Research objectives and motivation

Companies are expected to provide products and solutions to customers quickly and efficiently (Hänninen et al. 2012). There is an increased need for companies to address the challenges of the product development such as:

- More effective production of software
- Bringing new products to the market more quickly and efficiently
- Meeting customer and segment-specific needs more efficiently
- Getting new products to the market on time.

The effective production of software has been an issue in the mobile device industry for years. The background of this topic lies in the nature of the short life cycles of mobile devices and the demand for improved capability to bring new products faster and more efficiently to the market. If company misses a mobile device market window, this will impact seriously on the company’s productivity and market share. Successful product development is critically dependent on the consideration of stakeholders’ expectations and relationships throughout the
project life cycle (Bourne & Walker 2005, Cleland 1999). Stakeholders often have hidden power and influence, and they have a crucial impact on product development success (Bourne & Walker 2005, Yang et al. 2011).

Numerous stakeholders must be considered during software product development. Stakeholders are thought of congruently in the literature as persons, organizations, institutions, societies, neighborhoods, groups, and the natural environment (Mitchell et al. 1997). Today, stakeholders play a significant role in almost every project’s course of events (Karlsen 2002), and projects are influenced by many different stakeholders. Project stakeholders can be distinguished between stakeholders that are internal (intra-organizational) and external to the organization (Pirson & Malhotra 2007). The success of products is dependent on the ability to address stakeholders’ expectations (Bourne & Walker 2005). In order to create successful new mobile products, it is essential to identify stakeholders and understand how they interact with each other, and how they influence product development projects.

Software productization is a phase of a new product development project in which software functionalities and features of the new mobile device are finalized and integrated as a first commercial version of a software release to the market. Software productization takes place at the end of the mobile device development lifecycle. It is typically a phase during which a mass of different stakeholder demands and expectations, software and functionality related requirements, and change requests and technology expectations are met. In addition, the project is competing against time. The allocated resources are often exceeded, and costly and difficult decisions must be made when modifying and optimizing the content of the software release. The forthcoming software release target date creates increased stakeholder expectations to get as many features as possible included in the first software release package. Any new software features or functionalities added in the software productization phase increase the risk that the software product launch schedule targets are not met.

The objective of this thesis is to increase understanding of stakeholder interactions in cross-functional software productization in the context of mobile software development. Mobile software is an essential part of a mobile product. Mobile software productization is an integrative and finalizing phase of mobile product development. In practice, it is more complicated than that. At this stage, many stakeholder needs and new software requirements are still coming in from stakeholders, and balancing and trade-offs need to be accomplished. The development of a mobile software product is a cross-functional work effort
involving influential stakeholders who have interests and sometimes agendas and ambitions of their own. Considering and fulfilling the stakeholders’ needs is crucial in order to create successful software products. However, addressing all needs equally may not be a good approach, because the software productization may end up in a situation where the software product is overloaded with endless software requirements and, in other words, will never be mature enough for the market.

The motivation of this research is to provide theoretical and empirical insights into stakeholder interactions in cross-functional software productization in the context of mobile software development. The author’s long work experience in R&D is the second source of motivation. As is the case for many development companies, it is important to acknowledge stakeholder groups and the interactions between stakeholders in software productization. Finally, it is important to find ways to understand how these interactions could be managed in a more systematic way.

The objective has been derived from a mobile device company and the study is conducted in a large cross-functional R&D product development organization. The capability to accelerate mobile software product development and the ability to ensure the right timing to the market is in the interests of all of the company’s stakeholders and shareholders. Admitting that with embedded software development, the related hardware and systems are generally considered, they are left out in this research. Before the mobile software reaches the productization phase, numerous decisions must be made and much development work must be done in many organization units in the company. The decisions are evaluated from many perspectives, technological, cost, market pull and push, time-to-market and being the first in a particular technology area by weighting some criteria more than others. One viewpoint is that the software product needs to fit the choices and goals of the company.

The results of this research are also useful for managers, firstly, to provide visibility to the number and types of stakeholders and their interactions in software productization which are dealt with every day; and secondly, to increase awareness of the importance and impact of stakeholder interactions in software productization.
1.2 Research problem and questions

Research on interactions in cross-functional environments has been focused on studying cooperation and collaboration across selected functions in the context of new product development (Chen 2007, Fain et al. 2011, Fredericks 2005, Gemser & Leenders 2011, Holland et al. 2000, Kim & Kang 2008). This research aims to fill the gap caused by the lack of studies about stakeholder interactions in cross-functional software productization.

A mobile product consists of large software packages. For example, Nokia has reported that software development costs make up approximately 80% of the total mobile phone development costs (Jaakkola 2009). Mobile software product planning and development is a highly interactive process. Internal company stakeholders are involved in expressing their needs and expectations. During the development phase, hundreds of people participate in designing and engineering software components and making numerous decisions. The research questions can be stated as follows:

1. What stakeholder interactions exist in cross-functional software productization?
2. How do stakeholders associate through interactions in cross-functional software productization?
3. How could stakeholder interactions be coordinated in cross-functional software productization?

The research questions are related, but their focus is different. The research operates on a company level, but is valid even beyond the case company. The examination was conducted through a case study in a research company, and the phenomenon under study was investigated through soliciting the views of members of the case company’s software product development organization. Managerial implications were then derived from the research.

1.3 Research approach

This thesis is based on qualitative research involving a case study consisting of theoretical and the case company-specific sections. It is characteristic of qualitative research that the study is conducted in a real-life context, trying to make sense of or interpret the phenomena (Yin 2009). The case study approach fits particularly with this research, because the topic concerns social and complex
phenomena (Yin 2009) that occur in interactions. The case study is often used to understand the dynamics within a single scenario (Eisenhardt 1989), and this approach is adopted in this study, too. The research questions of this thesis focus on identifying stakeholder interactions from a cross-functional mobile software productization perspective and explaining the contextual impact of those interactions. As stated by Yin (2009), the case study method is useful in the research of contextual conditions.

The company-specific data was collected through open interviews, participating observation, and company documentation. This combined data collection method when using a case study approach is advocated by Eisenhardt (1989) and Yin (2009). The use of multiple sources converging the facts and findings increases the quality of the case study. The informants were selected among experienced senior managers of the case company to investigate their subjective perspectives. An overview of the research process and the major steps, according to Niiniluoto (1999), are presented in Figure 1.

![Research process overview and major steps](image-url)

**Fig. 1. Research process overview and major steps (based on Niiniluoto 1999).**
The research was carried out in parallel with the author’s work in the case company, dealing with real-life projects and in close cooperation with practitioners in product development projects in software productization situations. The theory section was carried out by reviewing the area of research through the means of analyzing related studies.

1.4 Prior literature review

This section provides an introduction to the extant literature carried out on topics relevant to the theoretical basis of this study. The structure of this literature review is composed as follows. Section 1.4.1 provides an overview of previous stakeholder studies relevant to this research in order to expose the concept itself, and the background of stakeholder thinking and how stakeholders may influence company and product development. Interaction-related research is visited in Section 1.4.2. Section 1.4.3 introduces studies in the productization field and Section 1.4.4 provides a view of mobile software development research.

1.4.1 Stakeholder research

Stakeholder theory has been a great topic of interest and has been studied from numerous perspectives since the 1980s. The pioneering work of stakeholder theory was introduced by Freeman (1984) in his book, *Strategic management: A stakeholder approach*, which described the idea of companies having stakeholders with characteristic features. Freeman’s essential study has been a basis for many other stakeholder scholars, and stakeholder theory has been applied in different business contexts (Fassin 2008, Friedman & Miles 2002, Mitchell et al. 1997, Post et al. 2002, Rowley 1997). Donaldson and Preston (1995) extended the stakeholder model to define and explain the concept of legitimate stakeholders, which should be seen as part of a company’s environment, having an impact on its activities.

The roots of stakeholder theory lie in organization theory and organizational strategy (Yläranta 2006). The primary idea of stakeholder theory is that the company has multiple stakeholder groups who it needs to operate with, balance and consider their interests and expectations (Freeman 1984, Jones & Wicks 1999). It has traditionally focused on two areas: 1) definition of the stakeholder concept (Freeman 1984); and 2) grouping and categorizing stakeholders in order to understand relationships between individual stakeholders (Rowley 1997).
Many stakeholder theory-related studies have concentrated on the identification, classification and management of stakeholders (Rowley 1997), trying to explain the interdependencies between the stakeholders and the organization (Yläranta 2006), as well as addressing organizational decision-making and the implementation of decisions (Donaldson & Preston 1995). Some researchers have integrated stakeholder theory with theories from disparate areas such as resource dependency theory, prospect theory, and organizational life cycle models, to further develop stakeholder theory (Andriof et al. 2002, Jawahar & McLaughlin 2001, Venkataraman 2002).

Stakeholder theory has been used from many perspectives. Social science stakeholder theory focuses on concepts of justice, equity and social rights, with a special perspective of stakeholders’ moral influence to projects and change initiatives (Gibson 2000). This approach is seen as very broad due to its dimensions varying from the business environment through social, physical, and quality influences (Bourne & Walker 2005). The instrumental stakeholder theory approach aims to maximize shareholder value on time via good stakeholder relationship management (Donaldson & Preston 1995). This view explains mutual stakeholders’ dependence through trust and commitment, and motivational forces (Bourne & Walker 2005). Normative stakeholder theory considers how managers or stakeholders should act and view the goals of organization, from an ethical perspective (Friedman & Miles 2006). Convergent stakeholder theory highlights project managers’ role in striving for mutual trust and cooperative relationship-building with stakeholders (Jones & Wicks 1999) and targeting actions based on ethical standards (Bourne & Walker 2005). The ethical and moral issues of stakeholder research arose during the 2000s (Gibson 2000, Lampe 2001), mainly addressing two problems: fair and efficient people management, and the social or moral responsibilities of the company (Orts & Strudler 2009).

The role of stakeholders or stakeholder groups is a focal point of stakeholder management theory. Stakeholder theory is one perspective of strategic management (Freeman 1984, Donaldson & Preston 1995). According to Freeman (1984), the major driving force mode of stakeholder management is active stakeholder involvement and creating the future together. Company strategy is an outcome of the process involving managers from different levels of the organization. Stakeholder management addresses the integration of processes and balancing stakeholders and issues via interactions (Yläranta 2006). The term ‘stakeholder’, as used nowadays by organizational and management researchers, can be defined as an actor with a two-way interaction or exchange of influence
with the corporation, as defined by Freeman (1984) and Carroll (1993). Mitchell 
et al. (1997) suggest a method for analyzing stakeholders and balancing their 
competing demands by looking at stakeholders through three factors—power, 
legitimacy, and urgency—which determine stakeholder salience by classifying 
stakeholders qualitatively by their possession or attributed possession impacting 
on the organization. Friedman and Miles (2006) introduce the view that an 
an organization should be seen as a set of groups of stakeholders, and the main 
purpose of an organization is to manage and consider those stakeholders’ needs 
and expectations.

The identification and classification of stakeholders enhances understanding 
and explaining the company and its environment (Aaltonen 2010), and broadens 
the vision of the company management (Mitchell et al. 1997). As stakeholders are 
seen as crucial for the success and performance of organizations, there has been 
an increasing need to understand and manage stakeholders more profoundly 

The concept of the stakeholder

Freeman’s (1984:46) definition, “A stakeholder in an organization is any group or 
individual who can affect or be affected by the achievement of the organization’s 
objectives”, is the most classic and commonly used (Clement 2005, Nilson & 
Fagerström 2006), and it is applied in this study, too. This definition has, however 
been changed over the years, even by the originator himself. Freeman (2002:41) 
redefined stakeholders as, “those groups who are vital to the survival and success 
of the corporation”. The popularity of stakeholder-related research has led to 
some confusion in the usage of the term ‘stakeholder’. Roberts and Mahoney 
(2004) found 125 stakeholder-related studies in 65% of which the term 
stakeholder was used without referring to any version of stakeholder theory.

Many scholars view stakeholders from a corporate perspective (de Bakker & 
den Hond 2008, Savage et al. 1991). This view has been complemented from an 
individual stakeholder perspective by Rowley (1997). Donaldson and Preston 
(1995) consider stakeholders as government, political groups, investors, 
customers, trade associations, communities, and employees. Stakeholders are vital 
for the company, and therefore it is essential to understand who is a stakeholder 
and how to assess stakeholders’ influence and impact on the organization in terms 
of their significance to the company’s economic interest. Stakeholders can be
divided into internal and external company stakeholders (Aaltonen & Sivonen 2009, Deetz 1995, Winch 2004) as shown in Figure 2.

![Fig. 2. External and internal company stakeholders (based on Deetz 1995).](image)

External stakeholders establish a framework for the company that affects its threats and opportunities. They are not formal members of the project community but can influence or be influenced by the project (Aaltonen & Sivonen 2009). Internal stakeholders constitute the daily operations of the company (Mark-Herbert & Schantz 2007) and are usually members of the project coalition (Winch 2004). Internal and external stakeholders are mutually dependent (de Bruijn & ten Heuvelhof 2002). Stakeholders can be grouped into primary and secondary stakeholders. Primary (real) stakeholders possess directly or contractually a real claim or interest in a company, having self-interest and concrete objectives. It is harmful for the organization’s survival if primary stakeholders are not considered continuously. Secondary stakeholders are considered to be more like pressure groups (Clarkson 1995).

It should be noted that stakeholders are heterogeneous (Harrison & Freeman 1999). Stakeholder groups may have multiple roles and interests (Winn 2001), and each stakeholder may have its own subsets with associated obligations and
influences (Fassin 2008). Often a stakeholder of one company may also be a stakeholder of another company with its own stakeholder network (Key 1999, Rowley 1997). The stakeholder groups of a company may be fairly diverse, but it is common for them to have shared joint interests as well as potential conflicts (Post et al. 2002). Savage et al. (1991) consider stakeholders to threaten or cooperate with the company by dividing them into claimants and influencers, as stakeholder groups have their own interests and valid needs, and they use their power to act according to their interest by, for example, influencing the companies’ products (Nilson & Fagerström 2006). Therefore, it is essential to identify and consider stakeholders and apply their expectations, needs and strategic goals (Jassawalla & Sashittal 1998, Yläranta 2006). The stakeholders with most to lose from the organization’s actions can be considered the most important, but this does not reflect their influence and power (Bourne & Walker 2005). Relationships between the company and stakeholders change over time, and the role of a stakeholder depends on the situation, time and on-going issues (Friedman & Miles 2002, Kochan & Rubinstein 2000, Phillips 2003, Winn 2001). Hence, the stakeholder perception of pressure, threats and opportunities varies in each life cycle stage of the company (Jawahar & McLaughlin 2001). As a summary, stakeholders can be simply described according to Dinsmore (1999) as “the ones who hold the beef”.

**Stakeholders’ impact on company performance**

A company can be considered as a complex system of stakeholders and nets of relationships (Clarkson 1991, Lim et al. 2005, Yläranta 2006). Freeman (1984) considers stakeholder management to be the responsibility of the company managers and as involving managing and integrating the relationships and interests between different stakeholder groups. This is a challenging task, because different stakeholders have different interests which may sometimes conflict with each other and the company goals (Freeman 1984, Orts & Strudler 2009). Stakeholder management is commonly seen as a key success factor and enabler for a company’s overall performance and success (Clarkson 1991, D’Aveni 1995, Hillman & Keim 2001, Joyner & Raiborn 2005).

Cleland (1999) provides a process with practical steps for stakeholder management. The major process steps are: identification of stakeholders; specifying the nature of the stakeholder’s interest; measuring the interest; predicting the stakeholder’s future behavior to satisfy them or their stake. The
final step is to evaluate the influence of a stakeholder’s behavior on the project. Bourne & Walker (2005) argue that this process may be seen as similar to project management practices such as planning, organizing, motivating, directing and controlling the project. Kolk and Pinkse (2006) propose fairly similar steps for stakeholder management by identifying three themes: identifying the nature of stakeholders; examining under which circumstances and how stakeholders influence organizational decisions and operations; and identifying different strategies to deal with stakeholders. The identification and analysis of stakeholders is often overlooked, although a systematic and analytical approach is clearly needed in order to understand who is a stakeholder, their claims and their mutual inter-relationships. Proper analysis is needed to involve important stakeholders effectively in decision-making (Reed et al. 2009).

Stakeholders play critical role as partners, consumers, owners, and employees, and ignoring them may lead to project failure. According to Clarkson (1991), companies which manage their stakeholder relationships proactively or aggressively have an above-average economic performance. In contrast, companies managing stakeholder relations reactively and defensively have a below-average economic performance. Kaplan and Norton (1992) developed the stakeholder management concept further by creating a tool for managers called the ‘Balanced Scorecard’, which can be used to examine the performance of the stakeholder groups of the company. Hillman and Keim (2001) focus on the connection of shareholder value, stakeholder management, and social issues in their studies. They found that investing in stakeholder management may complement shareholder value creation and lead to increased shareholder wealth. Stakeholders as a term are considered as the owners of the company, emphasizing the financial and performance aspects of the company in a time frame. Competitive advantage is based on developing stakeholder relationships which enable important resources and capabilities in the business operating environment.

The primary purpose of stakeholder management frameworks is to support managers in identifying the influencing stakeholders and their claims and capability of influencing decision-making in the company or organization. Stakeholder management can also be described as part of risk management. Project managers are in a key role to ensure and lead this work (Bourne & Walker 2005). Although stakeholder management has been studied widely, there are issues regarding how the demands of different stakeholders or stakeholder groups’ are integrated and how conflicts in their demands are managed (Yläranta 2006).
Criticism of stakeholder theory

Stakeholder theory has been commented on and criticized widely from a theoretical and philosophical point of view in academic literature (Donaldson and Preston 1995, Gibson 2000, Kaler 2003, Key 1999, Weiss 1995). Some scholars claim that there are misinterpretations regarding stakeholder theory, when it argues that all stakeholders should be taken into account equally, despite the fact that some stakeholders contribute more than others to the organization (Gioia 1999, Marcoux 2003, Phillips 2004). However, stakeholder management does not imply that executives have to direct equal amounts of attention to all of their constituents (Denchev & Heene 2003). Within the stakeholder categories, the level of attention and obligation may vary with each attribute operating in a continuum, or series of continua, rather than as a binary, present or absent term (Mitchell et al. 1997, Phillips 2003).

Additionally, it is claimed that the theory ignores the internal stakeholder heterogeneity (Harrison & Freeman 1999) and the fact that stakeholders may often belong to more than one stakeholder group (Jansson 2005), having simultaneous and multiple roles with varying interests (Freeman 1984, Pesqueux & Damak-Ayadi 2005, Post et al. 2002, Winn 2001). Pesqueux and Damak-Ayadi (2005) propose that the stakeholder’s position and the role being played at the given moment in time should be considered when analyzing their influence. Fassin (2008) argues that stakeholder theory has not been justified properly and does not reflect the complexity of the real world. It is important to notice that, for example, the stakeholder group called ‘employees’ may consist of managers, production personnel and blue- and white-collar workers representing different educational backgrounds and responsibility areas and having clashing interests and priorities with varying personal and group interests.

According to Fassin (2008), the company is considered as a central hub in Freeman’s stakeholder model, and as such it should be presented as a group of senior management of the company and not the firm itself. It ignores the fact that stakeholders of the company deal mainly with company management, whose responsibility is to look after shareholders’ interests. Winn (2001) describes the manager as both identifier and interpreter, and therefore the crucial mediator of stakeholder influence. The original Freeman model does not consider the interactions among stakeholders, even though in real life there are numerous multilateral contacts among stakeholders (Key 1999). Fassin (2008) extends the
expanded stakeholder model by including the interactions between stakeholder groups in the model.

1.4.2 Interactions

The company consists of cross-functional active participation and information exchange between different stakeholder groups. Most interaction-related research focuses on interactions between an organization and its external stakeholder groups (Freeman 1999, Mitchell et al. 1997, Rowley 1997). Studies of an organization’s internal interactions are often carried out under cross-functional themes such as communication (Kuhn 2008), collaboration (Dougherty 1992), and integration (Engelen et al. 2012, Fain et al. 2011). Aarikka-Stenroos and Jaakkola (2012) explored joint problem-solving processes and as a result developed a framework for co-value creation for more manageable and improved collaborative interaction processes. There is some confusion in the terminology of interactions in the literature. The terms interaction, collaboration and cooperation are often associated with the same phenomena in the context of cross-functional themes, but they have different definitions even though collaboration and cooperation are considered synonyms by some scholars.

Stakeholder management from a single focal organization’s viewpoint is very limited, and interactions within the project network of several organizations have not been considered widely (Aaltonen & Sivonen 2009). To be successful in today’s competitive and fast changing environment, it is widely recognized that information and knowledge should be managed and integrated into the functional operations of the company. This is achieved via social interactions, referring to the extent to which organizational members interact with each other in terms of trust, communication, and coordination (Chen & Huang 2007). Particularly when a company is promoting an innovative climate, this leads to an increased need to exchange creative ideas through social interaction across the company (Chen & Huang 2007, Norrgren & Schaller 1999). Interactions between stakeholders promote value creation for and with the company (Grönroos 2008, Payne et al. 2008).

The creation of new products is a cross-functional and multidisciplinary process, and consequently, companies put a great deal of effort into coordination work and trying to structure their organization in the most efficient way, aiming to increase functional interaction. Specialized functions and departments create difficulties in coordination, and the need for cross-functional integration and
cooperation increases (Frederics 2005, Gupta et al. 1986, Ruekert & Walker 1987). Cross-functional collaboration and interactions are the key performance factors in the company. According to Rowley (1997), dense networks consisting of shared information and expectations between stakeholders seem to establish unified stakeholder pressures, leading the firm towards conformity. In contrast, less dense networks undermine information exchange between stakeholders. Furthermore, Rowley (1997) suggests that organizations should not respond to each stakeholder individually but instead interact with the entire stakeholder network.

Scholars have emphasized the importance of the development and maintenance of the organization and its stakeholder relationships as desirable goals (Clarkson 1995, Donaldson & Preston 1995). Mitchell et al. (1997) formulate a theory of stakeholder identification and salience, providing an explanation for the nature and effects of stakeholders’ interactions in the organization and giving clarity to the academic discussion; that is, who is a stakeholder and how to identify the stakeholders, which should be paid attention. The theory predicts stakeholder salience to managers and a model for explaining and evaluating the impacts of stakeholder interactions upon the focal organization. Freeman and McVea (2001) argue that the identification of stakeholders and their interconnections is a critical step. Interfunctional interactions are important for the company for several reasons: to be able to develop products that meet customer needs, enabling a productive relationship between R&D and other functions (Cadogan et al. 2005, Kahn 2001). The role of functional integration during new product development, and the need and importance of collaboration between functions such as marketing, R&D and operations, vary over time (Olson et al. 2001).

Interaction-related conflicts between marketing and R&D have been studied during recent years (Massey & Kyriazis 2007, Olson et al. 2001, Ruekert & Walker 1987). Ruekert & Walker (1987) examined how and why marketing people interact with other functions, and as an outcome they developed a framework to be used for describing and understanding interactions between functional areas and different marketing positions. Cross-functional relationships between marketing and R&D attract added interest from scholars because this interface is often in conflict in terms of creating a new product (Atuahene-Gima & Evangelista 2000, Fain et al. 2011, Kahn 2001, Massey & Kyriazis 2006). It is acknowledged in the literature that marketing and R&D functions play a critical role in new product development, and the better these functions are coordinated
and integrated, the greater the chance for a successful new product (Massey & Kyriazis 2006). Conflicts between marketing and other functions are seen to lead to reduced market performance (Menon et al. 1997). Conflicts and uncertainties between functions may occur when R&D specialists focus on resolving technology related problems and issues, while marketers drive customer and market demands (Olson et al. 2001). An overview of interaction-related studies is presented in Table 1.

<table>
<thead>
<tr>
<th>Researcher</th>
<th>Research topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aarikka-Stenroos &amp; Jaakkola (2012)</td>
<td>Collaborative value co-creation in knowledge intensive business services</td>
</tr>
<tr>
<td>Ackerman &amp; Eden (2011)</td>
<td>The power and interests of stakeholders</td>
</tr>
<tr>
<td>Akbar &amp; Hassan (2010)</td>
<td>Collaborative interaction among software development team</td>
</tr>
<tr>
<td>Bourne &amp; Walker (2005)</td>
<td>Identification of stakeholders and visualizing their influence and power</td>
</tr>
<tr>
<td>Cadogan et al. 2005</td>
<td>Interfunctional connectedness and conflict in export marketing</td>
</tr>
<tr>
<td>Kahn 2001</td>
<td>Market orientation, interdepartmental integration and product development performance</td>
</tr>
<tr>
<td>Kuhn (2008)</td>
<td>Intra-organizational power and stakeholder relationships</td>
</tr>
<tr>
<td>Massey &amp; Kyriazis (2007)</td>
<td>Interpersonal trust and conflicts between R&amp;D and marketing managers during new product development project</td>
</tr>
<tr>
<td>Mitchell et al. (1997)</td>
<td>Theory of stakeholder identification and salience</td>
</tr>
<tr>
<td>Neville &amp; Menguc (2006)</td>
<td>Framework for understanding stakeholder multiplicity through strength, direction and synergies between stakeholders</td>
</tr>
<tr>
<td>Olson et al. (2001)</td>
<td>Cooperation between marketing, operations and R&amp;D and its implications for project performance</td>
</tr>
<tr>
<td>Ruekert &amp; Walker (1987)</td>
<td>Marketing interactions within intra-organizational functions</td>
</tr>
</tbody>
</table>

As stated earlier, considering stakeholder expectations is considered a key element for successful strategy implementation for the companies. Since projects need to interact with numerous stakeholders (Aaltonen & Sivonen 2009), it is essential to understand stakeholder dynamics and the intensity of their influence. Several methods have been developed to help to map the stakeholders. Bourne and Walker (2005) developed a method for identifying, mapping and visualizing stakeholders, their influence and power within the performing organization. Neville and Menguc (2006) introduce a framework to understand stakeholder multiplicity through strength, direction and synergies between interacting
stakeholders by establishing their studies on Mitchell’s *et al.* (1997) framework of stakeholder salience explaining the process of managerial decision-making. Akbar and Hassan (2010) study collaborative interaction among software development team and claim that the success of the project depends on how effective the collaboration and the interaction are among the team members and the client of the project. Interaction has been acknowledged to be an important factor for successful project team work, along with new collaborative working practices such as agile methods.

According to Ackerman and Eden (2011), management of the interfaces between the different and often competing expectations of company stakeholders is seen as one of the most important enablers for the successful implementation of strategic goals. They introduce a model in which those stakeholders important to a specific organizational situation and their significance are identified, and the interactions between them captured. All stakeholders of the organization are influenced by one stakeholder’s actions. One stakeholder’s power can be described in relation to their position in the network of other stakeholders. Ackerman and Eden (2011) call this power-interest grid a ‘Stakeholder Influence Network Diagram’, as shown in Figure 3.

![Stakeholder power-interest grid](image)

*Fig. 3. Stakeholder power-interest grid (based on Ackerman & Eden 2011).*
This diagram reveals stakeholder dynamics and both formal and informal relationships. In the model, the stakeholders are positioned in four categories. The upper two categories represent stakeholders with most interest in the organization. ‘Players’ are stakeholders that have a high degree of power to support or sabotage the company’s strategy, while ‘Subjects’ are interested but have less influence. ‘Crowd’ and ‘Context setters’ represent the lower categories and can be considered as potential stakeholders who may have power to influence in the future (Ackerman & Eden 2011).

1.4.3 Productization

The term ‘productization’ is not widely used in the literature (Artz et al. 2010), and there is no unique definition for the term either in the academic literature or in the English dictionary (Hänninen et al. 2012, Simula et al. 2008). According to Simula et al. (2008), two concepts in the literature have similarities with productization: mass customization and commercialization. These concepts are introduced in later sections. However, few varying definitions are available. Hoch et al. (1999) consider productization as a standardization of the elements in the offering. Hietala et al. (2004) explain the term productization as including several technological elements, from designing a product to selling and distributing the product. Some researchers consider productization as a development process starting from analyzing customer or market needs, designing the actual product and the capability to produce the product or service (Flamholz 1995). Myers et al. (2002) use the term productization as consisting of a list of variables to define the maturity of technology. Productization has been used in the context of product and service creation and being a key enabler for company success (Flamholz & Akselhirli 2000). Productization is considered as a process for producing innovations, including phases such as analyzing customer or market needs, product design, and creating a capability to develop the product (Valminen & Toivonen 2007). The term productization has also been used in the context of an attempt to define intangible services more clearly for more easily sellable products (Simula et al. 2008). Hänninen et al. (2012) present a concept of rapid productization as a framework for companies that want to fill the gap in their offering in their current product or service portfolio in order to respond rapidly to market needs. An overview of productization-related studies is given in Table 2.
Table 2. Overview of productization related studies.

<table>
<thead>
<tr>
<th>Researcher</th>
<th>Research topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artz et al. (2010)</td>
<td>The transformation process (productization process) from customer-specific software development to product software development</td>
</tr>
<tr>
<td>Flamholtz (1995)</td>
<td>Productization process as part of efficient product and service creation and delivery</td>
</tr>
<tr>
<td>Flamholtz &amp; Akselhirli (2000)</td>
<td>Product and service creation</td>
</tr>
<tr>
<td>Hietala et al. (2004)</td>
<td>Degree of product software and software service business productization</td>
</tr>
<tr>
<td>Hänninen et al. (2012)</td>
<td>Rapid productization preconditions and challenges</td>
</tr>
<tr>
<td>Lehtimäki et al. (2008)</td>
<td>Productization of innovative technology as productized offering</td>
</tr>
<tr>
<td>Myers et al. (2002)</td>
<td>Productization through a set of variables for defining the maturity stage of technology</td>
</tr>
<tr>
<td>Simula et al. (2008)</td>
<td>Productization framework</td>
</tr>
<tr>
<td>Valminen &amp; Tolonen (2007)</td>
<td>Productization of services</td>
</tr>
</tbody>
</table>

Productization is argued to be a predecessor for a product’s successful entry into the market, including production launch, and ramp-up activities, marketing materials and program development, supply chain management and training and support services (Lehtimäki et al. 2008). The concept emerged during the 1990s and can be seen as a natural step to take to optimize differentiating products and their variants. Productization is often linked to the concepts of mass customization and commercialization (Simula et al. 2008). These terms are now introduced briefly because of their similarities with productization.

**Mass customization**

No commonly accepted definition for the term ‘mass customization’ exists (Simula et al. 2008). Mass customization is used to provide products to different customer segments while maintaining productivity (Gilmore & Pine 1997, Hart 1995, Simula et al. 2008). Hart (1995) explains mass customization as a method to satisfy customer needs while focusing on mass production. This term combines the high volume generic products with the lowest cost of mass manufacturing with minimal product modification (Simula et al. 2008). Da Silveira et al. (2001) specify mass customization as a method for establishing an array of products where various customer needs can be fulfilled in a productive way. According to Grieves (2006), the difference between mass customization and traditional mass
production is that mass customization relies on customer-driven development. Generally speaking, mass customization differs from productization in that mass customization focuses on manufacturing and cost management (Simula et al. 2008).

**Commercialization**

Commercialization is typically seen as a last phase of the innovation or new product development process (Rosenberg 1988, Simula et al. 2008, Song & Montoya-Weiss 1998). Commercialization is a cross-functional process consisting of product development, production ramp-up, marketing, supply chain, sales, training and support (PDMA 2004, Simula et al. 2008). Commercialization is also often used to discuss how to bring novel technologies into a profit-making position when technology might already have some applications or is just in search of them (Simula et al. 2008). The goal of commercialization is to introduce a new product or technology to the market (Lehtimäki et al. 2008). Simula et al. (2008). As stated by Simula et al. (2008), the goals of mass customization and commercialization explain the major difference between these concepts.

**Software productization**

Software productization has been discussed mostly in the context of the development of standard and customized software products and services (Alajoutsijärvi et al. 2000, Artz et al. 2010, Flamholz & Aksehirli 2000, Jaakkola 2011, Valminen & Toivonen 2007). It has been traditional in the software industry for software to be seen as highly customizable, so that it can be tailored for specific customer needs (Codenie et al. 1997). A number of studies in the software productization field focus on software products that are traded on their own (Hietala et al. 2004), and little has been studied in terms of software productization when software is an essential part of a product and built into it. The software productization process emphasizes the change in the attitude of the company.

A recent definition of software focused productization is stated by Artz et al. (2010:7) as follows: Productization is “the process of transforming from customer specific software development to a standard software product”. A software product is defined as “a packaged configuration of software components or a software-based service, with auxiliary materials, which is released for and traded
in a specific market” by Xu & Brinkkemper (2005:3). Increased competition has
driven companies to seek new, more productive and optimized, methods of
software development. Productization is seen as a major prerequisite for the
continued growth of the software industry (Alajoutsijärvi et al. 2000). Recently, a
new role has been introduced in software business: the product manager. This role
reflects the strategy of the company to apply a process of moving from
customized software development to standardized software products (Artz et al.

In applying a software productization process, a company transforms from
customer-driven (customized software) towards market-driven product software
development (Artz et al. 2010). However, customers often need customized
software in order to integrate the software into certain situations (Hietala et al.
2004), so it is essential that the right timing and maturity of the company is
analyzed thoroughly before making this strategic change in its business (Artz et al.
2010).

Some other software product development strategy-related terms need to be
acknowledged. Commercial (or Common)-Off-The-Shelf (COTS) products can
be considered as software products or product families targeted for customers, as
they are without any customer-specific modifications. Tailored software products
are customer-specific solutions aimed at solving a customer’s special need or
problem. Modified-Off-The-Shelf (MOTS) software products are customizable
and modifiable to some degree, for example through customer-specific
parameters (Warsta 2001).

1.4.4 Mobile software development

The growth of the mobile device market has created a huge demand for mobile
applications. Mobile devices are not solely used for calls and messaging, but are
expected to correspond to desktop computers and to run a growing range of
software applications. Global mobile device manufacturers are also typically
software systems companies. The mobile software development industry is
extremely competitive and dynamic in nature (Abrahamsson et al. 2004, Lal et al.
2001), and under rapid expansion, leading to very short mobile product life cycles
(Bogner & Barr 2000, D’Aveni 1995, Zeidler et al. 2008). New technologies and
improved software development platforms enable better performance and more
complex applications for mobile users. A mobile device consists of hardware,
mechanical and software parts, and it is of the size of a mobile phone, containing

35
an information system (Lindholm et al. 2003) and in which the software is a crucial part (Abrahamsson 2005). Rapidly changing customer needs and hyper competition are typical of the mobile software industry (Riel & Lievens 2004). Under these circumstances, the right timing of new products and services is crucial for the company (Shen et al. 2000).

Because of the fast life cycle of mobile phones, the industry has been seeking solutions for getting mobile devices and software to the market more quickly. Mobile software development is characterized by special features such as short time to delivery, added difficulty in identifying stakeholders and their needs and expectations, great number of mobile variants with particular hardware characteristics, firmware and operating systems (Abrahamsson 2005), and increased complexity (Wasserman 2010). Mobile devices are continuously increasing in memory and power. Hayes (2003) remarks on the evolving bandwidth, coverage and security and inherent constraints (reduced data entry capability, memory capacity and power reserve).

Software organizations are currently investigating how to improve their business operations in technology, administration and product development practices and processes (Ahmed & Capretz 2007), and this is also the case in the mobile software development industry. Mobile software development organizations are urged to eliminate all excesses from their software engineering processes. Wasserman (2010:398) defines software engineering as follows: “Software engineering is a process by which an individual or team organizes and manages the creation of a software-intensive system, from concept through one or more formal releases”. Vainio et al. (2005) stress the need to integrate customers as stakeholders in different roles more tightly into the mobile software product development process. Zeidler et al. (2008) have studied the integrated product development process for mobile software and services. They emphasize clear product definition, the multidisciplinary background of the development community, and senior management commitment as key enablers for a successful new product development process.

Agile methods have become a widespread trend in mobile software development. The basis of agile methods lie in the principles of Lean Manufacturing, which originated in the 1940s, and Agile Manufacturing, which originated in the 1990s (Salo 2006). The applicability of agile methods in the context of mobile software development has interested many scholars (Abrahamsson 2005, Laanti 2012, Rahimian & Ramsin 2007, Wasserman 2010).
An overview of research in the mobile software development field is presented in Table 3.

Table 3. Overview of research topics in mobile software development.

<table>
<thead>
<tr>
<th>Researcher(s)</th>
<th>Research topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abrahamsson (2004, 2005)</td>
<td>Agile methods in mobile software development</td>
</tr>
<tr>
<td>Laanti (2012)</td>
<td>Agile methodology in large-scale mobile software development</td>
</tr>
<tr>
<td>Lindholm et al. 2003</td>
<td>Mobile usability</td>
</tr>
<tr>
<td>Rahimian &amp; Ramsin (2007)</td>
<td>Agile methodology in mobile software development</td>
</tr>
<tr>
<td>Riva &amp; Del Rosso (2003)</td>
<td>Software product family approach in mobile software development</td>
</tr>
<tr>
<td>Vainio et al. (2005)</td>
<td>Customer-based software development for mobile markets</td>
</tr>
<tr>
<td>Wasserman (2010)</td>
<td>Software engineering issues in mobile application development</td>
</tr>
<tr>
<td>Zeidler et al. (2008)</td>
<td>Mobile software development as part of integrated new product development process.</td>
</tr>
</tbody>
</table>

Mobile software development can be considered as large-scale software development (Laanti 2012), and as such very complicated, slow, expensive, and unpredictable by nature (Bosch & Bosch-Sijtsema 2010). There has been a drastic change in the field during the past decade. Mobile software development is no longer controlled by manufacturers or network operators. Newcomers such as Apple and Microsoft have had a significant impact on mobile device manufacturers and network operators as well as mobile software developers by imposing their own rules. The new world provides opportunities, but also constraints and threats, to mobile software development. The rapidly increasing number of mobile users and new advanced technology establish a huge entertainment market potential (Abrahamsson 2005).

Mobile manufacturers have tried to find more productive and efficient methods by applying software product line practices to utilizing their software assets. A software product line is applicable when software application development is based on common architecture and core assets (Ahmed & Capretz 2007, Nascimento et al. 2008). Riva and Del Rosso (2003) have examined the software product family approach in the context of mobile software development. Software product line thinking is clearly one way to accelerate mobile software deliveries and their quality, and to offer a higher throughput of mobile devices. Riva and Del Rosso (2003) estimate that most of the delays of embedded products are, however, because of delays with the software rather than hardware faults.
1.4.5 Conclusions of the literature review

The development of complex products, such as mobile devices, is influenced by multiple stakeholders with many diverse expectations and requirements. In order to create successful products, their claims need to be understood and balanced by product development projects. Given the central role of stakeholder interactions in the productization phase during product development, there are gaps in the previous literature.

While studies on stakeholders of product development or product management focus primarily on identification of stakeholders and mapping their influence on product development and the company’s overall performance, the theoretical base of how stakeholders interact during product development has been rather limited. Furthermore, little theory or empirical research exists on software productization and stakeholder interactions.

Consequently, the main objective of this thesis is to contribute to software productization by bringing valuable new information and theoretical understanding of how product development project stakeholders interact during a product development project and influence it in the context of mobile software development. In addition to contributing the product development project stakeholder interactions, research is seen to contribute to stakeholder research and the research of complex software product development projects by providing empirical evidence of stakeholders’ interactions.

The literature review identifies the need to enhance the understanding of stakeholder interactions and how they influence product development projects. The increased understanding broadens the product development and product management decision making process with regard to stakeholders. The previous literature demonstrates the lack of understanding of the complexity of stakeholder interactions even though software productization helps companies to produce solutions for diversified markets and customers more efficiently. Instead many studies focus on the identification of stakeholders and mapping their influence. Even through product development scholars have identified the importance of stakeholders’ role for successful products, the interactions between the product development project and stakeholders have not been studied in-depth.
2 Mobile software development

This chapter introduces the theoretical view of mobile software development which represents the context of this research.

2.1 Mobile market

According to Steven and Ariel (1996) the telecommunications industry entered an era of rapid change in the early 1970s. The decades since the 1980s have seen a vast growth in telecommunications services and products. Complex segmentation of consumer demands and rapid changes in the product market characterize the telecommunications industry (Fransman 2001). It is typical of the high technology market that there are periodic and unpredictable changes which result in total change in the competitive positions in the industry, with new leaders often emerging. In other words, the competition in the high technology market is extremely fierce. Business success is achieved through customer satisfaction accomplished by the introduction of new products and product differentiation. (Pleatsikas & Teece 2001). The match of market and technology is a fundamental concern because it is a central element in creating the expected future cash flow that accelerates shareholder value (Bond & Houston 2003).

The worldwide penetration of mobile devices is explosive (Zeidler et al. 2008, Zhang & Prybutok 2005). A mobile device refers to a product which is about the size of a mobile phone, including the information system (Lindholm et al. 2003), and the software plays a major part in the product (Abrahamsson 2005). Competition in the worldwide mobile market is aggressive and increasingly dynamic (Vainio et al. 2005, Zhang & Prybutok 2005). The mobile industry is considered a high-technology industry that differs from more stable industries in terms of emphasizing performance rather than price based competition (Pleatsikas & Teece 2001).

There are 6.8 billion mobile subscriptions worldwide in February 2013. The increase is huge: in 2010, there were 5.4 billion subscribers, and in 2011, there were 6.0 billion subscribers (mobiThinking homepage 2014). The numbers of worldwide mobile subscriptions in 2013 are divided geographically as shown in Table 4.
Table 4. Mobile subscribers worldwide by region, 2013 (based on mobiThinking 2014 homepage).

<table>
<thead>
<tr>
<th>Region</th>
<th>Mobile cellular subscriptions (millions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>6,835</td>
</tr>
<tr>
<td>Developed nations</td>
<td>1,600</td>
</tr>
<tr>
<td>Developing nations</td>
<td>5,235</td>
</tr>
<tr>
<td>Africa</td>
<td>545</td>
</tr>
<tr>
<td>Arab States</td>
<td>396</td>
</tr>
<tr>
<td>Asia &amp; Pacific</td>
<td>3,547</td>
</tr>
<tr>
<td>Commonwealth of Independent Stated (CIS)</td>
<td>476</td>
</tr>
<tr>
<td>Europe</td>
<td>790</td>
</tr>
<tr>
<td>The Americas</td>
<td>1,048</td>
</tr>
</tbody>
</table>

The market contains a huge amount of information for the mobile industry (Heurese et al. 2012) and establishes a continuous source of new opportunities (Zeidler et al. 2008). According to the Portio Research homepage (March 2013) an estimated 1.2 billion people worldwide were using mobile applications in 2012. Estimates of mobile application users by regions and the forecast for 2017 are shown in Table 5, demonstrating the vast opportunities in mobile software products.

Table 5. Users of mobile applications worldwide by region (based on Portio Research March 2013 homepage).

<table>
<thead>
<tr>
<th>Region</th>
<th>2012</th>
<th>2013 (estimate %)</th>
<th>2017 (estimate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>1.2 billion</td>
<td>N/A*</td>
<td>4.4 billion</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>0.4</td>
<td>32%</td>
<td>2.2</td>
</tr>
<tr>
<td>Europe</td>
<td>0.3</td>
<td>28%</td>
<td>0.9</td>
</tr>
<tr>
<td>North America</td>
<td>0.2</td>
<td>17%</td>
<td>0.4</td>
</tr>
<tr>
<td>Middle East &amp; Africa</td>
<td>0.2</td>
<td>13%</td>
<td>0.5</td>
</tr>
<tr>
<td>Latin America</td>
<td>0.12</td>
<td>10%</td>
<td>0.4</td>
</tr>
</tbody>
</table>

*Figures for 2013 not available, regional percentages are estimates

In order to gain market and competitive advance, companies need to gather and integrate information from many sources, such as competitors, consumers, market, end users and authorities (Vainio et al. 2005). Marketing represents the customer’s voice (Fein et al. 2011) by gathering, exploring and forwarding new product requirements to the company. The integration between marketing and new product development is seen as one of the key enablers of successful new product development (Barczak 1995, Souder 1988).
There is a strong correlate of new product development and communication between marketing and R&D. Communication is considered crucial for successful product development (Cooper 1984, Dougherty 1992, Griffin and Hauser 1996). The market needs and opportunities view represents the sources of information for the mobile business (Heureuse et al. 2012). The information gathered from the market should be analyzed and integrated during the product development process in order to answer customer needs and to create successful products for the market. Market pull represents unsatisfied customers and their new needs (Brem & Voigt 2009). In order to create successful products, the company must satisfy the markets’ and customers’ demands (Fain et al. 2011, Song & Thieme 2006).

A market opportunity can be seen as the intersection of the company capability and customer need. Companies apply existing and incremental technologies to address market and customer needs but new technologies or emerging customer needs may create new market opportunities. (Bond & Houston 2003). Communication between marketing and product engineering is considered a key factor for successful products (Dougherty 1992, Griffin & Hauser 1996, Souder 1988). Market information should be evaluated and responded to by new product development project communities (Adams et al. 1998, Frishammar & Hörte 2005). The relationships between marketing managers and R&D are managerially important, because the current flat organization structures require cross-functional cooperation among individuals who have no hierarchical control over each other (Massey & Kyriazis 2006, Williams 2001).

2.2 Characteristics of mobile software

The technology progress of mobile devices is fast and their computation capabilities are growing with the demand for specialized software for these devices (Rahimian & Ramsin 2007). Mobile software is a non-scientific term which is used to describe the context of this research, and which can designate any software used in portable devices. It can be described through particular characteristics and through how it differs from traditional software as it has many constraints and special requirements that must be considered, such as:

- Wireless communication (issues with availability and disconnection, bandwidth availability, heterogeneous networks, security risks)
- Mobility issues (address migration, location dependent information)
- Portability issues
Various standards, protocols and network technologies
Limited capabilities of mobile devices (low power, small sized user interface, low storage)
Strict time to market requirements
Special privacy and customizability needs. (Rahimian & Ramsin 2007:337).

Mobile products provide an important channel for personal communications and an increasing number of innovative services for users (Sugai 2007). A software engineering is defined by IEEE (1990) as “the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software”. The mobile software industry faces environmental particularities such as: a high level of competitiveness (Lal et al. 2001); many different operating systems and legal issues (Zeidler et al. 2008); added difficulty in identifying stakeholders and their requirements; fast-paced development and short time-to-delivery (Abrahamsson et al. 2004), but it has similarities with any other embedded software engineering (Wasserman 2010). The mobile software development environment consists of dynamic surroundings in which frequent and fast-paced modifications, needs and expectations must be dealt with (Abrahamsson 2005). While software-related studies are available in the literature, the characteristics of mobile software or mobile service development have not been analyzed widely (Abrahamsson 2005, Zeidler et al. 2008). Abrahamsson (2005) has studied mobile device game development companies and generalized his observations to other mobile development domains. He argues that internal and external factors can be considered as characteristics of mobile software development. External factors are those over which development organizations do not have control but just need to adopt themselves, e.g. mobile device manufacturers, network operators, end users, non-standardized technology, and distribution channels. Factors that a development organization can control are the diversity of competence and skills, adequacy of testing and software architecture.

Technical development and functionality are often focused on mobile product development, and the holistic user experience of the end user is omitted because of the management team’s demand to get the product to the market rapidly. However, this approach leads to frustrated and disappointed customers and company personnel through low demand and high support costs. End users’ contribution to the development process must be recognized to ensure a successful product (Vainio et al. 2005, Zeidler et al. 2008). Typical mobile software issues and constraints include integration of hardware, security reliability,
performance, and storage limitations (Wasserman 2010). According to Bosch and Bosch-Sijtsema (2010), there are currently three major trends in the mobile software development field:

1. The software product line approach, widely taken up by the mobile software industry over the last decade (Bosch 2002, Bosch & Bosch-Sijtsema 2010, Clements & Northrop 2001). The main goal of this approach is to try to benefit intra-organizational reuse of software assets e.g. targeting, to promote decreased development cost and faster time to market. The basis of this approach lies in creating a software platform line which is shared by a group of products. Several new products are launched and released per year, and the per product R&D expenditure can be decreased. Focusing on software product lines rather than products increases the complexity and organizational dependencies of software development (Bosch & Bosch-Sijtsema 2010).

2. Broad globalization of software development. Global mobile software companies have distributed development across many continents, elevating the degree of difficulty of dependency management dramatically (Bosch & Bosch-Sijtsema 2010).

3. Mobile software coupling with mobile ecosystems. A software ecosystem is defined as follows: “A software ecosystem consists of a software platform, a set of internal and external developers and a community of domain experts in service to a community of users that compose relevant solution elements to satisfy their needs” (Bosch & Bosch-Sijtsema 2010:68). The usage of third party developers is continuously increasing and is the acknowledged strategy of many mobile manufacturers. This however, adds to the complexity of software development due to dependencies between third party developers and the mobile company (Bosch & Bosch-Sijtsema 2010).

These major mobile software development trends are discussed in the following sections.

**2.2.1 Software product line approach**

Increased speed of product development is seen as the most beneficial advantage to using the product line approach (Muffatto & Roveda 2000). The software industry has also recognized the potential of using software product lines (Bosch & Bosch-Sijtsema 2010, Böckle et al. 2004). The adoption of software product lines is considered a key factor for improving quality, productivity and reducing
costs of software development in intra-organizational software engineering (Bosch 2002, Nascimento et al. 2008) and improving business operations and product development processes (Ahmed & Capretz 2007). A software product line consists of a product line architecture, including a set of reusable software components and shared assets (Bosch 2001). Clements and Northrop (2001:5) define a software product line as “a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission, and that are developed from a common set of core assets in a prescribed way”.

Mobile software products are often planned as product families in order to improve quality and productivity and reduce the costs of developing individual product variants by reusing the software (Nascimento et al. 2008) across a product platform in which a set of products shares the common assets (Robertson & Ulrich 1998). A single product family requires a shared common technology (Sanderson & Uzumeri 1995) and a shared platform used by all business units (Bosch & Bosch-Sijtsema 2010). Robertson & Ulrich (1998:20) define a platform as “a collection of assets that are shared by a set of products”. Product functionality is planned at a high level during the front end phase of concepting, demanding a great deal of coordination between different functions (Vainio et al. 2005). Although all mobile manufacturers create mobile products from a standard platform consisting of software components, each component can be used, extended or changed by different product teams. Each product team is able to select and integrate software components into a product. The software architecture plays a critical role in enabling the compositional approach. The clear benefit of this approach is that product creation teams have great independence. The downside is the risk of inefficiency of the system because of the possible duplicated extensions of components, and too many product-specific solutions for shared software components (Bosch & Bosch-Sijtsema 2010).

2.2.2 Distributed software development

The globalization of the software industry has become a popular business model for software organizations (Jaakkola 2009). The term ‘global software development’ has been studied widely by many scholars (Conchúir 2010, Damian et al. 2003, Herbsleb et al. 2000, Herbsleb & Moitra 2001, Herbsleb et al. 2005, Mockus & Herbsleb 2001). Globally distributed software development exists when development teams are separated globally and collaborate towards a
common goal (Carmel 1997). Global software development spans separate countries, continents, and time zones (Conchuir 2010).

Technology, the search for efficiency, and financial benefits as well as the possibility of finding skilled developers have been the major stimulants for distributed software development (Herbsleb & Moitra 2001). One source of motivation has been the attempt to utilize market opportunities (Cho 2007). However, despite the engineering and information systems supporting global collaboration, software projects are often delayed and do not work as they were intended to (Mann 2002). Past experience shows that despite the potential benefits, distribution brings many challenges and inefficiencies. Herbsleb and Moitra (2001) present strong evidence that multisite software development projects take longer than co-located projects because of delays in communication and coordination. They also note that distributed development does not need to be global to face the same problems. Espinosa et al. (2007) continue further, stating that distributed software development needs a remarkable amount of coordination work, because the software is developed simultaneously by many teams but will eventually be integrated into one product.

Global distribution of software development impacts significantly on the collaboration and communication of development teams. Tacit knowledge is important for working global teams and is seen to be a success factor for achieving organizational change. In new business strategy changes, part of the existing tacit knowledge needs to be changed, too (Nonaka 1994a). As tacit knowledge is implicit (Nonaka 1994a), it is difficult to make the needed change (Bosch & Bosch-Sijtsema 2010). Contextual awareness and working practices are difficult to share among globally distributed teams. Time zone challenges impact on accomplishing tasks. In addition, the coordination of global tasks needs more time (Powell et al. 2004). Effective communication becomes more difficult because of language and cultural differences (Kraut et al. 1999). For example Indian and Chinese resources are often used by companies as remote sites for coding, but the software or system architect or head engineer may be located in Europe. This may cause ineffective and slow communication and eventually lead to de-motivation and frustration thanks to a lack of autonomy and end responsibility at the remote sites (Bosch & Bosch-Sijtsema 2010). The mobile software business is more and more networked due to technological coalitions, partnerships and industry forums. This impacts directly on mobile software development because of the increasing number and variety of contributing stakeholders (Vainio et al. 2005).
2.2.3 Open software ecosystems

A software ecosystem (Bosch & Bosch-Sijtsema 2010) consists of interrelated hardware, software products, development community users, software organizations, industry, governments, and many other actors (Messerschmitt & Szyperski 2003). Software is no longer used alone, but interacts with many other software and hardware products (Yu 2012). According to recent trends, successful software product lines are transforming into platforms open for external developers to create their own solutions, products, and services on top of the software platform created by the mobile company. Mobile companies have clear responsibility over mobile product quality and safety, and that is why external entities are not provided with easy access to the actual platform. A true open ecosystem development model should allow unconstrained releasing of components by the owner of the platform, certified third party developers and other community members providing new features, services, and functionalities (Bosch & Bosch-Sijtsema 2010). Mobile users can download applications onto their devices from the portals.

In their transition from product-centric to product line approach software development, software ecosystems lead to new dependencies and a need for improved coordination between the associated organizations, development teams and software components. This coordination is needed throughout the product life cycle. External developers may have different views of the priorities for the platform functionalities and features. In architectural planning, the interfaces should be developed together with external developers and communities. In the platform release validation phase, the external communities must be involved to avoid breaks in platform functionality. However, the open software ecosystem approach seems to be the next step on from release train type development. A strategic decision is needed from the company to open the software platform to externals and even better consider and involve their customers when developing new products (Bosch & Bosch-Sijtsema 2010).

2.3 Mobile software product development

Product development is the phase in which the new product is designed and developed (Cooper 1993). Krishnan and Ulrich (2001) define industrial product development as exploiting market needs and opportunities commercially by utilizing technological opportunities with new products. New product
development can be considered as a set of activities which are carried out in transforming new product ideas into new product designs (Chiesa et al. 1996). New product development can be seen as a sequence of interlinked information processing tasks in which knowledge of customer requirements and needs is turned into product design (Meybodi 2003). Researchers and industries see new product development as an integrated process having many compromises between time to market, production cost, development cost and delivered customer benefits (Henard & Szymanski 2001, Smith & Reinertsen 1998). Most companies consider new product development as an end-to-end process including marketing, engineering, manufacturing and organizational development (Ulrich & Eppinger 2004). Companies’ success in the market depends on how effectively and efficiently they are able to create new products (Cooper 1993). The capability of creating new products is enabled by good integration and mechanisms to coordinate different functions (Ulrich & Eppinger 2004).

New product development performance can be improved through cross-functional integration elements such as overlapping and interacting product development phases (Gerwin and Barrowman 2002). The capability of launching new products successfully is based on idea generation and evaluation processes as well as monitoring own and other industries and current trends that reveal new opportunities. All new technologies and trends enabling products must be scored, evaluated and analyzed against the current portfolio, whether the new product is a complementary product or a substitute for an existing product(s). New product decisions are based on economic impact, value and revenue streams (Zeidler et al. 2008). A formal product development process improves the utilization of information and increases decision-making efficiency (Chiesa et al. 1996). Gupta and Wilemon (1996) mention key challenges of new product development including monitoring market developments and maintaining a spirit of inquiry. They also discovered that companies are struggling to maintain and accelerate the performance of R&D, and commercialization competences and capabilities in a turbulent environment in which the evolution of new technologies is immense. Smith & Reinertsen (1998) stress the market orientation towards customer satisfaction and needs being key factors in accelerating the time to market. New product creation happens in dynamic and turbulent environment (Doz & Kosonen 2008). The turbulence is caused by such changing factors as markets and customers, technology, competition, regulations, organization, globalization, and environmental changes (Kettunen 2009).
The mobile software industry is becoming more and more linked with numerous industry forums, partnerships and technology coalitions impacting directly on software product development because of the increasing number of contributing stakeholders involved in product development (Vainio et al. 2005). Xu and Brinkkemper (2005:525) define a software product as, “a packaged configuration of software components or a software-based service, with auxiliary materials, which are released for and traded in a specific market”. The main factors causing inefficiency and complexities in mobile software product development lie in the inter-team and inter-component processes. Inefficiency is often caused by engineering processes not being aligned with the platform and the system. A disconnection between engineering and business processes, e.g. the product management process, may cause extreme inefficiency in mobile software development (Bosch & Bosch-Sijtsema 2010).

Mobile devices are more and more standardized commodity products. New mobile device development is driven by consumer demand and fueled by technological opportunities. Stakeholders in the mobile product market have an important role in influencing the product, and they are increasing in number. It is still difficult to identify the most influential stakeholders, how they relate to each other and when they should be involved in different stages of the software product development (Vainio et al. 2005). It is important that stakeholders at all levels create, communicate, and share a holistic picture of the functional and economic view of the mobile software or application. Idea iterations should be a repetitive process in an innovative company, ensuring new product development that is technology-driven and trend-oriented (Zeidler et al. 2008).

According to researchers, the integration of marketing and R&D is a key factor for creating successful products and services (Barczak 1995, Vainio et al. 2005). Customers expect improved performance and versatile mobile applications (Singh & Palmieri 2011). According to Vainio et al. (2005), the user’s role in mobile product development has been elevated to the top level and the value perceived by users through user experience is a focal point in new mobile product planning; however, there is still no proper procedure for thoroughly managing internal or external stakeholder information. The major challenges for new product development are monitoring market development effectively, performing R&D, developing commercialized technological capabilities, and fostering alliances as well as accelerating the development and commercialization of new products (Gupta et al. 1986, Vainio et al. 2005). Companies have tried to manage mobile software products by adopting the software product line approach over the
last decade (Bosch & Bosch-Sijtsema 2010, Clements & Northrop 2001). A product portfolio presents all product families defining the base product (member of a certain family) which becomes a reference for the rest of the product family, while a product map includes information about product capabilities (Nascimento et al. 2008).

2.4 Stakeholders of software product development

Product development projects interact with a number of stakeholders. In order to understand stakeholders’ power and interests, they should be analyzed by identifying, categorizing and differentiating them (Freeman 1984). New product development depends on trust between different functions (Bosch-Sijtsema & Postma 2009, Bstieler 2006), informal interactions (Tsai & Ghoshal 1998), and shared objectives (Xie et al. 2003). Considering the needs and requirements of different functions is part of project management (Aaltonen & Sivonen 2009, Olander & Landin 2005). Successful projects follow the steps of identification, analysis and management of stakeholders (Miller & Olleros 2001); however, it is important to understand key stakeholders’ relative influence and expectations (Bourne 2006). PMI (2004:24) describe stakeholders as individuals and organizations “that are actively involved in the project, or whose interests may be positively or negatively affected as a result of project execution or successful project completion”. The integration of stakeholders in the context of project management as an enabling factor for successful projects has been studied widely (Aaltonen 2010, Bourne & Walker 2005, Olander & Landin 2005). Artto and Wikström (2005:349) state that “projects are part of overall business and a central part of the development, strategic sight and maintaining of the firm’s competitivenes”. Addressing multiple stakeholders’ expectations is a very difficult task and very little is known about how stakeholders should be managed in the context of new product development (Driessen & Hillebrand 2010).

There is no single model to describe or define which functional areas impact on new product development (Aleixo & Tenera 2009, Jassawalla & Sashittal 1998, Sherman et al. 2005, Song et al. 1997, Ulrich & Eppinger 2004). Marketing, design and manufacturing are seen as central functions of new product development (Ulrich & Eppinger 2004), as Song et al. (1997) wants to add marketing and R&D, production and finance functions. The success of project outcome is critically dependent on relationship and management skills to achieve the project objectives by addressing stakeholder expectations (Cleland 1999).
Walker (2003) has illustrated a stakeholder model of a project, as shown in Figure 4.

Fig. 4. Stakeholder model for a project (based on Walker 2003).

The product development process is considered as an unknown set of actions (i.e. a ‘black box’) by many researchers (Aleixo & Tenera 2009, Ulrich & Eppinger 2004). The following typical phases can be identified: concept development, product design, and pilot production/testing (Schroeder 2003). Ulrich and Eppinger (2004) see product design as being the focal point of new product development and argue that new products are developed through concept development, system-level design, detailed design, testing and refinement, ending with production and ramp-up. Boer (1999) identifies five stages in product development: Beginning with Raw Ideas, Conceptual Project Stage, Feasibility stage, Development stage, and finally Early Commercialization stage.

Software product managers face many requirements originating from different stakeholders such as the company board, research & innovation, development, support, services, and sales & marketing (Weerd et al. 2006a). The organizational structure of a company reflects the division of work according to functional expertise such as research and development, marketing and production, and this increased need to coordinate the interdependencies (Ainamo 2007). Functional structures based on one function dominating over the other functions tend to produce better outcomes for less innovative products, such as those involving lien extensions or product improvements (Astley & Zajac 1991). Organizations coordinate inter-functional interactions across organizational activities related to the new product development process and across the
functional and participative structures. Participative structures based on work organized across functions are likely to improve effectiveness (Ainamo 2007). Stakeholders of software product development are addressed in the following sections. Key stakeholder functions are marked in italics (e.g. gathering, sharing).

2.4.1 Marketing

The market is an enormous source of information for any business (Heureuse et al. 2012). Market orientation is about gathering, sharing, and using information about the market in order to make decisions (Jantunen & Smolander 2006). Hurley and Hult (1998) emphasize the influence of organizational market and learning orientation as prerequisites for successfully adopting or implementing new ideas, processes or products. The high-technology industries differ from more stable and mature industries due to their emphasis on performance- rather than price-based competition. The degree of differentiation of products is high. The product differentiation of new innovative products is one of the key elements in successful business and customer satisfaction. The mobile industry is unpredictable, having continuous shifts in which dominant technologies and trends are undermined, leading to total change in the competitive position in the market and emerging new market leaders (Pleatsikas & Teece 2001).

The marketing function supplies the voice of the customer (Fein et al. 2011) and as such the first contact with customer and market needs to gather the new product requirements and channel them forward to be processed in the company (Weerd et al. 2006a). Cooperation between R&D and marketing has a significant impact on new product development project success (Driessen & Hillebrand 2010, Ernst 2002). Marketing researchers often see the R&D function as subordinate, and R&D considers marketing as a static or even limited function, leading to gaps between these functions (Fain et al. 2011, Gupta & Wilemon 1986). The dissemination of market information is based on coordination mechanisms to ensure that people involved in new product development projects get the relevant information (Griffin and Hauser 1996, De Luca and Atuahene-Gima 2007). Market information should be evaluated and responded to by new product development project communities (Adams et al. 1998, Frishammar & Hörte 2005). The relationships between marketing managers and R&D are managerially important, because the current flat organization structures require cross-functional cooperation among individuals who have no hierarchical control over each other (Massey & Kyriazis 2006, Williams 2001). A product development project
typically needs integration between R&D and the marketing function setting the new product objectives and product features, and resolving cost-design-performance trade-offs (Griffin & Hauser 1996). People from R&D and marketing tend to differ in terms of their education and professional backgrounds, leading to difficulties in achieving efficient coordination (Gupta et al. 1986). The new product development process as such easily causes considerable conflicts between R&D and marketing due to conflicting priorities, goals and objectives (Dawes & Massey 2005, Gupta & Wilemon 1986, Massey & Dawes 2007, Song et al. 2000).

Studies indicate that conflicts between R&D and marketing are more the rule than the exception (Holland et al. 2000, Massey & Kyriazis 2006, Moeanart & Souder 1990). However, it should be noted that conflicts are not always harmful (Holland et al. 2000). Menon et al. (1997) recognize two types of conflict: dysfunctional conflict with unhealthy behaviors characterized by withholding information, distrust and hostility during interactions; and functional, healthy conflict with vigorous challenge of ideas, assumptions and beliefs. Menon et al. (1997) conclude that functional conflicts may be beneficial and dysfunctional conflicts are harmful. The personnel of the marketing function often take a coordinative role in which they link needs and requirements outside the company with other functional units within their own company (Ruekert & Walker 1987). For a company developing products trying to meet customers’ expectations and needs, the interactions between R&D and other departments are crucial (Kahn 2001). Management of the interactions of marketing and other functions provide a good basis for companies to be more adaptive, efficient and effective (Krohmer et al. 2002).

Marketing departments’ drivers for inter-functional interactions are the desire to achieve company targets and business unit level objectives, as well as personal goals. Interactions in the company are driven mainly by the same goal, but different individual objectives are sources of conflict (Ruekert & Walker 1987). Product development needs information about the market and technology. Marketing acts in a boundary-spanning role in the sense that marketing people are aware of customers’ needs and their purchasing behavior (Li & Atuahene-Gima 2001). Product managers define the product strategy and product roadmaps; tear them down into product releases and business requirements explaining those to development teams (Botzenhardt et al. 2011). Product managers are expected to plan the go-to-market and ensure product support (Kittlaus & Clough 2010, Weerd 2006b). The market is seen as an abstract stakeholder from a product
development project perspective, standing for potential customers, competitors and analysts (Weerd et al. 2006a). Market pull can be characterized as “inadequate satisfaction of customer needs which results in new demands for problem solving” (Brem & Voigt 2009:355). The market represents potential customers, competitors, and analysts such as Gartner (Weerd et al. 2006a). A successful new product development process responds to market demands with a right product. Marketing represents the customers’ voice, while R&D has the capability to create the product (Fain et al. 2011, Song & Thieme 2006). The customer is the source of new feature requests, often during the process of closing the deal or when they are already using the product. These requests may find their way to the product manager directly, but also through functions like services, support, and sales & marketing (Weerd et al. 2006a).

Communication between marketing and product engineering is seen as one of the most important factors in the success of new product development (Dougherty 1992, Griffin & Hauser 1996, Souder 1988). Communication with customers allows firms to learn from customers and make the right products to satisfy their needs. The mobile device business is considered as market driven, as it is characterized as offering superior solutions and experience, focusing on customer value, converting customer satisfaction into loyalty, driving to energize employees, moving to intimate market understanding, and leveraging brands assets (Day 1998). Technology innovations are pushed to the market initiated in the company’s R&D via the marketing function. Market pull is an expression of market need, and is a trigger and sign for R&D to fulfil the need (Martin 1994).

2.4.2 Corporate management

The highest authority in the company, which consists of company board members, undertakes corporate management. According to research, the role of corporate management is critical for successful new product development (Cooper & Kleinschmidt 1995). The role and the actions of corporate management are emphasized in providing an environment in which cross-functional collaboration is promoted and dysfunctional state is avoided (Souder 1981). Corporate management has discretion over the decisions (Griffin & Hauser 1996, Leenders & Wierenga 2002, Ruekert & Walker 1987) and that may undermine the knowledge change between different functions and the effectiveness of the collaboration (De Clercq et al. 2011).
Corporate management is liable for the definition and communication of the company’s strategy, vision and mission, and managerial supervision of different functions (Weerd et al. 2006a). Corporate management must give a high priority to and define a clear overall strategy for new product development (Griffin 1997, Holland et al. 2000, Jassawalla & Sashittal 1998). Hutt et al. (1995) argue that a fundamental task for management is to create a strategy map with which organizational members can strongly identify. Cooper and Kleinschmidt (1995) emphasize the critical importance of corporate management, acting as executive sponsors of projects and maintaining commitment. Corporate management may communicate some of its requirements through strategy as technology drivers or directly to product managers (Weerd et al. 2006a).

The majority of the research literature indicates that senior managers’ involvement has a positive impact on single project success (Young & Jordan 2008, Zwikael 2008). In contrast, some researchers have found that senior managers’ involvement may also have negative impacts on project success (Kessler 2000, Bonner et al. 2002). Jonas (2010) mentions that senior managers’ involvement may have simultaneous positive and negative impacts as value-creating and value-destroying events. In critical situations, senior managers tend to delay and hesitate to make a decision to abort a project which they have strongly supported, even in the case when it can be clearly seen that continuing the project induces more damage than value (Bonner et al. 2002, Ernst 2002, Jonas 2010). Strongly committed senior managers may end up in a political conflict in which they compete with each other, and this leads to longer project development times (Zirger & Hartley 1996). Furthermore, senior managers may overrule agreed processes by interfering with new product development, causing more distrust and weakening cross-functional cooperation (Jonas 2010). When managers bypass the processes, relationship-based role conflicts may arise (Jehn & Mannix 2001). A formal process for governing organizational decisions promoting fairness (Li et al. 2007) mitigates poor cooperation and increases trust and confidence in the organization (Jehn & Mannix 2001, Jonas 2010).

Senior managers are the starting point of the company image, atmosphere and tolerance for failure, therefore they have a central role in defining the vision of the product. If the definition is done outside the development project, most likely there will be re-reviews and rework and the project will get sidetracked. It is difficult to involve senior managers in the early phase of product development, because they may avoid and feel uncomfortable communicating their vision and talking technology to lower-level product creators. The project manager is often
under an obligation to create a product and ask senior management for approval, which is like playing roulette. When problems occur, senior management may get overly involved, slowing the project by expanding decision-making loops and making them more complex and time consuming (Smith 1990). For the project development group to be able to move fast, the project manager’s authority should be acknowledged and should exceed the authority of the functional unit heads, otherwise the development happens through committee-thinking decision-making and becomes too slow (Larson & Gobeli 1988, Smith 1990). The development of the product needs a lot of communication and detailed, efficient and reliable methods of decision-making (Smith & Reinertsen 1998).

### 2.4.3 Software product management

The software market has shifted from developing customized software to standard software, thus the importance of software product management has emerged in the software industry. It is a fundamental and critical strategic value in many software companies (Weerd et al. 2006a). Software product management is defined as “the discipline and role, which governs the product (or solution, or service) from its inception to the market or customer delivery in order to generate biggest possible value to the business” (Ebert 2007:850). Software product management is a process which unites business and technology perspectives in software development in order to contribute customer value. In industries in which the software is part of a product, this may often be seen as a cost factor rather than adding value to the customers (Maglyas 2013).

Software product management is the well-organized processing of issues around managing requirements, and defining software releases and software products in an environment in which there are many both internal and external stakeholders (Vlaanderen et al. 2009, Weerd et al. 2006a). Product management is complex, since it includes requirement management, release management and deliveries and launches. Weerd et al. (2006a) propose that software product management has the overall responsibility for release planning, requirement management and product launches according to the framework shown in Figure 5.
Fig. 5. Software product management framework (based on Weerd et al. 2006a).

The software product management framework consists of four key software product management functions: 1) portfolio management, 2) product roadmapping, 3) requirement management, and 4) release planning, including internal and external stakeholders relevant to product development. It should be noted, however, that traditional product development cannot be applied directly to software products, as a software product differs from other products in the sense that there are no additional manufacturing or distribution costs of producing extra copies of the product (Botzenhardt et al. 2011, Cusumano 2004).

The complexity of product management is caused by interactions with stakeholders (Weerd et al. 2006a). Ettlie (2006) presents that companies which put effort into managing interfaces can gain considerable competitive advantages through their product development and utilizing their innovations. There is strong evidence that marketing and R&D managers are the most important decision-makers during new product development (Massey & Kyriazis 2007) because the success of the new product is more likely when these managers cooperate (Maltz & Kohli 2000, Souder 1981). For example, productized software is a result of complex interlinked elements (McLeod & MacDonnell 2011) in continuous interaction (Kautz & Nielsen 2004).
The mobile industry has a very fast tempo by nature. Quick business strategic changes are made at fairly short notice, and it is not only the R&D of the organization that needs to implement the change, but the change must happen throughout the whole organization (Bosch & Bosch-Sijtsema 2010). A well-established development strategy combined with development processes increases new product development performance and eventually company performance (Barczak 1995, Thomke & Reinertsen 1998). Agile approaches such as Scrum have been introduced in software development during the last decade as an answer to the rapidly changing environment and requirements. Scrum consist of set of rules, roles and artifacts stating how a team or a project should work. The rules of Scrum define how team project planning and team meetings, reviews, iterations and reviews and lessons learned should be carried out and collected. One of the central artifacts of Scrum is a Product backlog containing user stories, i.e. high level product requirements and a Sprint backlog containing tasks relevant to a software product under development. (Laanti 2012, Schwaber 2004). Agile requirements refining has been applied to the management of large requirements which are structured into smaller granularity (Vlaanderen et al. 2009). Agile software development methods are seen to bridge business and technology competencies and implement user-centered design towards a more systematic interaction between design and technology (Botzenhardt et al. 2011).

Many scholars claim that the traditional elements of successful projects such as meeting budget, time and performance goals are no longer sufficient for achieving a company’s objectives (Cleland 1999, Patanakul et al. 2012, Shenhar 2001). Global competition has brought the need for cross-functional teams aiming to better address global market needs, increase competencies and discover new markets (Ghobadi 2011). Many studies are available concerning cross-functional new product development between R&D and marketing (Atuahene-Gima 2005).

The literature defines a cross-functional team as a team consisting of individual representatives from various functions - departments, functional units, groups and organizations - who have knowledge and skills relevant to the project (Holland et al. 2000, Witt et al. 2001). Cross-functional teams and their performance have been studied widely and still interest many scholars (Holland 2000, Lu 2004, Mat & Jantan 2009). Many researchers have used cross-functional approaches in their studies in the context of new product development (Atuahene-Gima 2005, Brettel et al. 2011, Fain et al. 2011, Kim & Kang 2008).

Cross-functional methods are explained through measures of interactions and collaboration between functions and departments (Kahn 1996). Advantages of
cross-functional integration include increased communication and information flow (Randolf & Posner 1992). According to Sethi (2000), these elements are cornerstones for establishing a common product view, and as such should be considered as critical success factors in new product development. However, some downsides have been acknowledged, such as complex decision-making, especially in large teams, and lower efficiency and speed (Botzenhardt et al. 2011, Olson et al. 1995). Autonomy in decision-making in the context of new product development refers to the team’s freedom and autonomy to make their own project-related decisions, without being interfered with by senior managers outside the project team (Sethi & Sethi 2009). Low autonomy (centralized decision-making) urges functional managers to carry out function-specific tasks with no interest in leveraging relationships and ideas between different functions (Devinney 1995, Sethi & Sethi 2009). Decentralization takes the decision-making down to the managers who are close to the action level (Schminke et al. 2000), where they need autonomy in decision-making in order to be able to react and adapt to changing needs and situations (Lyonski et al. 1995). Empowering the cross-functional team increases their commitment and involvement in the work (Chen 2007); the frequency of communication is increased and contentious but constructive discussions are likely to evolve, resulting in widely accepted decisions (Yap et al. 1998). The key functions involved in software product management as stated by Weerd et al. (2006a) in their framework and shown in Figure 5 are described in more detail in the following sections.

**Portfolio management**

Business strategy defines the way in which companies compete in the market (Ruekert & Walker 1987). Portfolio management is seen as an activity in which a group of projects competing for resources are managed by portfolio management (Archer & Ghasemzadeh 1999, Dye & Pennypacker 2002). The product portfolio includes the whole set of products of the company. Product portfolio management is about creating a process for evaluating and selecting new projects and existing ones (Archer & Ghasemzadeh 1999). Products have a set of releases in the past, the present and for the future. Every release consists of either technical or functional requirements. Beckers et al. (2008) state that there are large gaps in high-level product management (product portfolio and product roadmapping) activities. It is crucial for companies to introduce timely and right new products to the right market to achieve medium and long-term success (Artz et al. 2010).
Corporate management sets very ambitious objectives for the near and long-term future (Chao & Kavadias 2008). Fast product development may inflate the costs of products and increase R&D expenses as well as degrade product quality or performance when managers make product development-related decisions and trade-offs but lack a true understanding of what is being given up in the trade (Smith 1990). According to Cooper et al. (1999), portfolio management and the prioritization of new products and R&D projects is critical because:

- It is about making strategic choices for the company
- It is about shaping the company’s future for years to come
- It is about resource allocation of marketing, operations and engineering functions
- It is about balancing resources between multiple projects

Managing product development activities through the portfolio is crucial for the company’s long-term success (Vähäniitty 2003). Project portfolio management is seen as an essential competence in companies which manage numerous projects simultaneously (Jonas 2010, Killen et al. 2008, Martinsuo & Lehtonen 2007). It is common for there to be no process for managing high-level management activities such as portfolio planning and roadmapping (Beckers et al. 2008), thus a proactive company’s product and project landscape management is important (Jonas 2010). Ebert (2009) has found that a vague product vision and strategy is one of the main root causes of insufficient product management. Portfolio management is about making decisions about the selection of products (Artz et al. 2010), and involves following market trends and opportunities, deciding the product development strategy, decision-making about product lifecycles, and managing product-related partnerships and contracts (Weerd et al. 2006a).

Product lifecycle management involves managing the processes needed to manage documentation, data and resources during the product lifecycle (Abramovici & Sieg 2002). One key activity of portfolio management is to define the product software lines including the products which share a common software platform to reduce development costs and the time-to-market (Pohl et al. 2005).

Attributes of the management role, such as significance, competence and clarity, are linked with the execution of managerial tasks aiming at cross-project coordination in a multi-project environment (Jonas 2010). The role of the portfolio manager varies depending on their assignment and authorization, being more like administrative type or strong shapers influencing the whole company’s
future (Blomquist & Müller 2006). A portfolio manager can be seen as a central co ordinative unit having an expertise in project portfolio practices (Jonas 2010).

Roadmapping

Product roadmapping focuses on long-term planning of new product releases and upgrades mapping those for identified or assumed market windows (Artz et al. 2010). Roadmapping is an approach that companies apply in software product business bridging the gap between business and product development. Bridging is crucial because the requirements for future software product releases need to be selected based on business decisions. In practice, this is a very challenging task (Damian & Zowghi 2003, Lehtola & Kauppinen 2006). Product roadmapping plays an important role in combining both product positions and development aspects as well as it connects the management, marketing and product development functions (Vähäniitty et al. 2002). A technology roadmap supports long-term technology management and planning activities, and shows its benefits when companies are exploring and sketching out organizational objectives, technological opportunities and resources (Phaal et al. 2004), and technology and market trends (Weerd et al. 2006a). Roadmapping is a method to build and manage the long term high-level development plan linking business and customer needs (Lehtola et al. 2005). According to Bosch (2000), roadmapping is often missed because of lack of experience, suitable processes and prioritization, and because of time-to-market pressures. The output of roadmapping can be used in high level content definition for future products. The organization must make strategic decisions about their future products and the domains in which they will be created, considering the market and technology trends (Artz et al. 2010).

Managers’ role in roadmapping is to make decisions considering time-scaled product positioning and product development (Artz et al. 2010) and to look for new business and product opportunities from market pull and technology push aspects (Phaal et al. 2004). This is enabled by creating a technology roadmap which enables the linking of the market pull and technology push domains and supports product roadmapping (Artz et al. 2010, Phaal et al. 2004). Roadmaps should provide a strategic path and visibility of products via key themes of products (Artz et al. 2010), and core assets of the products which are components shared within items from the same product line (Weerd et al. 2006a).

Roadmapping is a widely used technique in the manufacturing industry (Lehtola et al. 2007, Phaal et al. 2001). The method is rather new in the software
industry and the benefits of long term product planning are not widely acknowledged (Lehtola et al. 2007). Roadmapping is supported by a method such as GQM+Strategies. The method was introduced to align and link software related activities with corporate strategies and goals to target increased customer satisfaction in software intensive products and services. It applies measurement principles in linking the goals and strategies on different levels of an organization. The measurement approach is based on the Goal/Question/Metric (GGM) paradigm (Caldiera et al. 1994). The ultimate goal of GQM+Strategies method is to increase customer satisfaction in software intensive products and services (Münch et al. 2013).

Release planning and requirement management

Product development release planning is a critical and complicated process for the company (Artz et al. 2010). In particular, requirement organization may be very complicated (Weerd et al. 2006a). Release planning is about deciding what to include in future software release(s) of a software product (Colares et al. 2009). Defining requirements and functionalities for the next software release is difficult because the many different stakeholders have their own expectations and interests (Akker et al. 2005, Artz et al. 2010).

A software company must be able to select the most valuable requirements for their software releases (Lehtola et al. 2005), in order to be able to provide customers with maximal added value (Boehm 2003). The importance of right selection of features in the software product industry has been acknowledged (Penny 2002). Prioritization and the selection of features are the fundamental issues in release planning, i.e. which customers get what features and when. There are often interdependencies between top-priority and low-priority requirements that need to be implemented, and this increases the complexity and may lead to stakeholder conflicts (Carlshamre 2002). Carlshamre (2002) provides deep insight into release and requirement management activities and tools but does not give any guidelines as to who should be performing these activities in a large software company. Ruhe and Saliu (2005) state that there are three key roles who should attend software product and software release planning: project managers, stakeholders and the support environment.

Requirement management involves activities such as initiating, analyzing, prioritizing and maintaining a repository of requirements. Mobile software products are not targeted at a single customer, but at the whole market, and this
sets a necessity for managing customer needs and wishes efficiently and easily. The prioritization of requirements and decisions about feature content is focused on stakeholders, not one particular customer. In the productization phase, the requirements are gathered from the entire market segment (Artz et al. 2010). Software product companies must place importance on the selection and prioritization of requirements (Ebert 2005), and the priorities should be evaluated and reprioritized on a regular basis after considering changes in market conditions, customer needs, and business evolution (Wiegers 2003).

2.4.4 Platform development

In order to maintain competitiveness, companies are urged to find new ways for faster development (Ratamäki 2004). A platform can be defined as a “set of common components, modules, or parts from which a stream of derivative products can be efficiently created and launched” (Meyer & Lehnerd 1997:7). The platform approach creates a basis for aggressive market strategies for companies because of the resulting cost and time reductions in developing new derivative products (Muffatto & Roveda 2000). In order to survive in a competitive market, companies must try to cut new product development time (Cooper 2000, Ratamäki 2004, Smith & Reinertsen 1998). Getting one product to the market fast is not enough: several products must be launched into different market segments (Ratamäki 2004, Robertson & Ulrich 1998). This is particularly typical of high-technology products (McGrath 1995). High-technology companies are in a situation in which products should be developed faster, but at the same time products are becoming increasingly complex, and prices are decreasing (McGrath 1995, Ratamäki 2004).

The development of a new product consists of R&D activities (Cooper 1995, Ratamäki 2004), hence the R&D department is responsible for the execution of these activities and the release plans (Weerd et al. 2006a). The term Research and Development (R&D) is explained as an organizational function which performs research and development activities and activities themselves (Ratamäki 2004). The R&D function creates a product with differential competitive advantage by using the company’s assets and capabilities (Fain et al. 2011). R&D organization, software architecture and engineering processes are in deep interaction with each other. This should establish a basis for how R&D organization is structured. However, the processes and product or platform architectures are often defined by the organizational structure, leading to a situation in which the implementation of
the company’s business strategy is difficult and constraints are faced. In particular, when the company aspires to growth, strategy collisions may happen between the process and architectural choices and the existing organizational structure. The collision may happen when the R&D organization is still releasing products and platforms at the same time as there is a need to make the transformation into a new business strategy. There might be varying expectations as to the level at which the new business strategy alignment should be done. Although architecture is in a central role of software compositionality or optimally defined, it may not be fully utilized due to a lack of proper processes and implementation. Both formal and informal engineering processes need deep collaboration between individuals and teams (Bosch & Bosch-Sijtsema 2010).

The product platform strategy was introduced by McGrath (1995). McGrath (1995:39) defines the product platform as “a collection of common technical elements, especially the underlying core technology, implemented across a range of products”. The concept of the product platform was introduced in the information technology and software industry during the 1990s (Ratamäki 2004, Sääksjärvi 2002). In the software industry, hardware and software are referred to as platforms. The product platform is a common technological base for multiple derivative products, enabling efficient and effective product development. The process platform includes producing, sourcing and assembling the components and materials into a product as well as the final testing of the deliveries (Ratamäki 2004).

There is evidence that platform development brings cost savings compared to single product development because of the more productive and efficient product development (Meyer & Zack 1996). The product platform consists of building the technologies, organizational capabilities and consumer insights for product families targeted to different market segments or niches inside the segments (Meyer & Utterback 1993, Ratamäki 2004). When applying the product platform approach, a company must understand and forecast possible customer needs years ahead. The future technological selections create the basis for derivative products and product lines. The benefit of product platform thinking is that product differentiation can be done at a very late phase of product development. The increased productivity of the platform approach can be gained though making the product differentiation as late as possible (Ratamäki 2004).

It should be noted that the product platform is not an end product for the market itself, as the implementation of new product or service development is often undertaken in development projects with more accurate schedules and
objectives (Ulrich & Eppinger 2004, Välimaa et al. 1994). The interface between platform development and product development is essential, because the product platform transforms its output to product and service development projects to gain the desired derivative end products (Evans & Lindsay 2002, Rathamäki 2004).

2.4.5 Product development

High-technology companies aim to establish competitive advantages (Hung et al. 2007) through the creation of innovations (Aleixo & Tenera 2009). Organizations establish projects to achieve their business goals in the most efficient way (Blomqvist & Müller 2006). Projects can be considered as “temporary organizations within organizations” (Yap & Souder 1994). The major success factor for product development projects is when the company can capture customer needs, assess and transform them into the product specifications and more detailed requirements (Cooper 1995, Cooper 2000). According to Kärkkäinen et al. (2001), companies seem to fail at customer need evaluation and assessment, and this often leads to changes in product development projects. Project strategy can be defined as the direction the project follows, contributing to success within its own environment (Artto et al. 2008). Cusumano and Yoffie (1998) stress the importance of close cooperation and communication between the software developers and product platform development.

It has only been recently that the importance of stakeholder management in project development has been truly considered (Artto & Kujala 2008, Olander 2007, Olander & Landin 2005). Sutterfield et al. (2006) identify areas to focus on when pursuing overall project performance: continually communicating the project vision, mission and objectives, having proper project sponsorship established, proper planning, and managing conflicts and reducing resistance to change when new approaches are confronted. Although the importance of stakeholder impact on project success has been acknowledged, the research is still lacking both empirical and theoretical evidence of phenomena in the project stakeholder-related context (Aaltonen 2010, Achterkamp & Vos 2008).

Cross-functional product development

The increasingly competitive environment with the globalization of the market and fast-changing technologies has encouraged companies to use cross-functional methods in developing new products (Chen 2007, Kim & Kang 2008). A cross-
A functional team involves a group of people from various company functions such as engineering, manufacturing, R&D and marketing bringing together their different expertise, views and disciplines (Chen 2007, Lovelace et al. 2001), turning ideas into profitable products (Massey & Kyriazis 2007). According to Molin-Juustila (2006), cross-functional structuring is a common way to coordinate the work in a company by enabling the powerful coordination and management of interrelationships within one unit, but by contrast creating challenges in coordinating work between units. From a company perspective, a product development effort is often a disjointed activity confronting organizational units, e.g. marketing, engineering and management, and because of these divergent objectives and views, the project deliverable may lead to disappointments (Shapiro 1997). These operations create boundaries between functional units (Ainamo 2007). A development or project team consists of a group of individuals working towards a common goal for a limited time with their work coordinated through a project plan (Molin-Juustila 2006). Organizational groups are interdependent and need other groups for creating services or products. This interdependence creates a need to coordinate their efforts, and to consider each other’s needs (Knippenberg 2002). Cohen and Bailey (1997:241) define a cross-functional team as follows: “A team is a collection of individuals who are interdependent in their tasks, who share responsibility of outcomes, who see themselves and who are seen by others as an intact social entity embedded in one or more larger social systems (for example business units or the corporation), and who manage their relationships across organizational boundaries”. A cross-functional product development team is defined by Ancona and Caldwell (1992) as members of different departments and disciplines brought together under one manager and given the charge to make development decisions and enlist support for them throughout the organization.

Separated functions seem to emerge from functional boundaries and different knowledge domains, which lead to social barriers and reduces willingness and opportunities in information-sharing across functions (Kraut & Streeter 1995). Cross-functional collaboration is not always effective because of conflicts and misunderstandings, poor group cohesiveness, distracted or dissatisfied team members, and elusive agreements and consensus (Chaudron 1995, Donellon 1996, Swamidass & Aldridge 1996). Cross-functional methods are seen to have both positive (Lutz 1994) and negative (Donellon 1996) impacts on project performance. The performance of a cross-functional team consists of its ability to coordinate functional diversity, team leader characteristics, roles of conflict, and
roles of communication (Chen 2007, Keller 2001, Webber 2002). Within one cross-functional team, the members may have different views concerning the new product and its requirements (Molin-Juustila 2006).

The project manager takes a central role in leading and coordinating the functional decisions of relatively autonomous functional units and work (Ainamo 2007, Olson et al. 1995). A project leader who is trying to promote good intragroup collaboration is usually closer to one group more than the others, for example, being a member of one group, and therefore neutrality is difficult to sustain. The project leader can be considered as a boundary bridger, having significant interactions with outgroup individuals and other groups and facilitating and managing interactions between organizational groups (Hogg et al. 2012). The leader’s boundary bridging activity can be associated with lower conflicts and increased intragroup productivity (Richter et al. 2006). Effective intergroup leadership is needed to prevent disruptive conflicts and to realize the benefits of intergroup collaboration (Hogg et al. 2012). Intergroup leadership can be described as “leadership of collaborative efforts of more than one formal group or organization toward a joint goal” (Hogg et al. 2012:234). It can also be conceptualized through collaborative intergroup performance (Richter et al. 2006). Productive intergroup cooperation is not self-evident, and effective cooperation and work coordination capabilities between organizational groups are important for high performance (Hogg et al. 2012). It is essential for project success that project managers have the analytical and intuitive skills to identify the stakeholders of their project and understand and consider their influence and expectations, maximize their positive input and minimize harmful impact. Stakeholders should be engaged in providing the hidden positive power through interaction between individuals and their social networks. The more large and complex the organization, the more these analytical techniques are needed to ensure project success. Even if project managers identify stakeholder groups’ impact and influence on project development, it should be noted that they rarely have any control over the actions of external stakeholders (Bourne & Walker 2005).

Project management

Alexander (2005) suggests a conceptual model for a stakeholder taxonomy in order to identify and validate stakeholder roles. The key idea of a stakeholder taxonomy is to identify and list each stakeholder and their key drivers and
expectations in order to help the project manager to avoid missing the key demands of stakeholders and hence reduce risk and uncertainty during development. Danger signals may also be sent out by especially senior stakeholders which could be considered as early warnings about potential problems, and should be noticed by a project manager (Bourne & Walker 2005). These signals include interference without consultation, poor or missing communication links, too many reporting levels between the senior stakeholders, unfounded promises or commitments (Boddy & Buchanan 1999). The project manager is seen as a key role in coordinating and understanding stakeholders’ expectations and needs. In turn, project managers who are not able to react to stakeholders’ claims are seen to be the reason for poor project success or failure (Aaltonen 2010, Bourne 2006).

Project management has also been recognized as a key business process enabling the implementation of value delivery systems in companies (Jamieson & Morris 2004, Srivannaboon 2006). Project management is considered a special type of management to accomplish strategies, business goals and tasks within a defined resource frame (Srivannaboon 2006). Project managers focus on operational success in getting things done on time rather than business success, and this ‘operational mindset’ reflects the common understanding in the project management literature in which time, budget and performance are used to evaluate project success (Dvir et al. 2006). These traditional indicators may be misleading, in the sense that customer needs and requirements are not met (Atkinson 1999, Dvir et al. 2006).

According to Koskela and Howell (2002), no explicit theory for project management exists: they argue that the foundation of project management is espoused by PMI (2004). Furthermore, they claim that project management consists of two theories: a theory of project and a theory of management. A project is explained as the transformation of inputs and outputs managed according to numerous principles which suggest decomposing the transformation into smaller tasks and trying to minimize task-specific costs independently (Koskela & Howell 2002). Shenhar (2001) argues based on empirical studies that project management styles correlate with the technological uncertainty and system complexity of a project. He describes the characteristics of high-technology projects with high technological uncertainty as opposed to low uncertainty as follows: more design cycles, later design freeze, more intensity, and more informal communication. The success of a project depends on the level of interaction within the project team, but success can be gained only if the
project and organizational environment are supportive for individuals in the team and all members understand the team and organizational dynamics (Thamhain 2004). One major prerequisite for good performance is that project team members must have very clearly defined goals (Gallstedt 2003). Distributed multisite software development teams have become common as firms become global (Shrivastava & Date 2010). Along with opportunities such as leveraging resources and drawing advantages from markets and customers (Battin et al. 2001), increased number of defects, cost overruns, hazy understanding and differences in interpretations of requirements have been reported (Damian & Zowghi 2003, Herbsleb et al. 2005).
3 Productization

This chapter introduces the theoretical view of productization, as it is used in this research.

3.1 The concept of productization

In order to understand what productization is, the term ‘product’ should be clarified first. A product is traditionally considered as a physical manufactured artifact, but it may also be a non-physical function or performance offered to someone. There is no one unambiguous definition for the term ‘productization’ (Artz et al. 2010). Flamholtz and Hua (2002) consider productization as one of the six strategic building blocks for organizational success. They determine the fundamental idea of productization as building something to become a commercial product. It can also represent a well and clearly defined solution to a customer’s problem. The broadest view of productization describes it as a process or series of steps starting from product or service planning through development/manufacturing and ending with marketing activities (Flamholtz & Aksehirli 2000; Hietala et al. 2004). Productization as a process usually consists of components such as defining, specifying and profiling products, and modifying company know-how and capabilities to meet customer needs (Hänninen et al. 2012; Sipilä 1996). In simple terms, the goal of productization is to create the offering, technology or service so that customers are able to clearly understand what they are buying (Hänninen et al. 2012; Suominen et al. 2009). In order to be able to fulfill market and customer needs during productization, a company must understand those needs. Hänninen et al. (2012) and Simula et al. (2008) see productization as a concept supporting communication between marketing and design: after a company has been able to bring products into the supply system, it starts to look for means for better profits and efficiency. From a marketing perspective, the purpose of productization is to create a more appealing offering to the market. To summarize, a product can be defined from a marketing and sales point of view as an entity offered to the market or customer to satisfy a certain need (Hänninen et al. 2012; Kotler 2003).

A cross-functional integration approach is widely used in industrialized development processes. The product or service is created by multiple divisions of work with specialized competencies (Botzenhardt et al. 2011). Productization can be seen as the final integration phase to get the product to the market (Brettel et al. 2012).
Botzenhardt et al. (2011) explain cross-functional productization as an integration phase of product development. The integration phase has been studied intensively in the marketing and innovation management fields (Griffin & Hauser 1992, 1996, Gupta et al. 1986, Olson et al. 1995).


Productization of services

The most important motivation for productization of services is to increase efficiency, profitability, and competitiveness (Valminen & Toivonen 2007) in service business companies, such as consulting or software services businesses, in order to transform the service offering from an intangible into a more concrete product-like tangible offering, aiming to maximize customer needs and satisfaction and company profitability (Jaakkola 2011, Myers & Avison 2002, Simula et al. 2008, Valminen & Toivonen 2007). In other words, it is crucial for customer-driven development to be emphasized. A productized service may be a major differentiation factor compared to competitors (Bitner et al. 2008). Productization is sometimes related to a set of activities to create a single offering package including delivered features, documentation, training material and related services (Hietala et al. 2004, Jaakkola 2011). In addition, productization of services helps companies’ marketing activities through increasing understanding of the concrete offering that they want to market (Valminen & Toivonen 2007). Productization clarifies the content of the offering, which makes it easier for customers to select the product (Jaakkola 2011, Sipilä 1996, Valminen & Toivonen 2007).

Jaakkola (2011) perceives productization as a managerial practice in small service companies and identifies three key practices in service productization: 1) specifying and standardizing the service offering, 2) tangibilizing and concretizing the service offering and professional expertise, and 3) systemizing and standardizing processes and methods. Jaakkola (2011) combines management and productization by highlighting the management responsibility for balancing product and business strategy alignment in software product creation.
Productization of software products

From a software product business point of view, Artz et al. (2010:7) describe productization as follows: “the process of transforming from customer specific software development to a standard software product”, and they introduce a detailed process through productization stages for a company in order to make the change from customized software development to a standard software product with a customer focus. This definition considers software productization as a concretizing phase in which the service or software product offering will be finalized for the target market.

Artz et al. (2010) argue that a product can be called productized even if there is a small customized layer added to the software. This also implies the possibility of attaching new (customized) functionalities to the product software so that it can be integrated within existing environments. The main focus of the software productization phase is planning an optimal subset of requirements and considering resources in order to increase market share. The requirements collection and software content definition process is driven by market pull and technology push. Successful release planning involves the stakeholders selecting and validating the content of the software product. A set of core properties and features can be identified in all software products (Crawford 1997, Simula et al. 2008).

New directions of productization

The success of a product is determined by how well the insights from customer behavior can be linked to product development. As value-based feedback is needed from customers in software product development, accordingly decision making should be based on empirical experience gathered from experiments. Experimentation helps to evaluate, predict, understand, control and improve both the software development process and the product. Customer value can be defined as ‘whatever the customer is willing to pay for’. Value based feedback can be acquired through minimum viable product (MVP), which is not a complete product but which provides characteristics in order to discover the customer value. In practice MVP can be conducted in a process in which the product development team explores the concept design of the product in incremental cycles in collaboration with the customer. (Münch et al. 2013).
Lean Startup strives to ensure customers are addressed by the developers (Fagerholm et al. 2014) and the feedback loop between customer and product development is shortened (Ries 2011). In terms of Lean Startup methodology a value hypothesis and a MVP need to be developed. A MVP is considered an experiment and as such a first product consisting of a value hypothesis to test ‘whether a product or service really delivers value to customers once they are using it’. Lean Startup methodology aims to shorten the feedback loop from customers to product development. (Ries 2011: 67). New enhancements and features are developed and deployed continuously with software intensive products. This increases the need for developers to explore continuously what customers need and want through direct feedback and observation of their behavior. The product development team should identify and solve problems relevantly for customers and in that way deliver value for them. In order to create a capability for continuous learning from real time customers, companies are proposing to establish a development system in which the software functions are organized in an R&D experiment system. To achieve this transition the product management function must be involved and close collaboration with the lead customer is required for development of the product concepts. (Fagerholm et al. 2014).

As new needs and solutions are discovered and requirements are changing, the product backlog becomes larger. The product backlog should reflect reality so it is under constant change. The role of the product owner is essential for software product-related decision making and product backlog management in collaboration with the product development team. The product owner defines and prioritizes reviews and revises the high level value hypothesis, i.e. experiments. (Münch et al. 2013). The management of the backlog is also referred to as grooming. It is an ongoing process in which details and estimates are added and the order of items in the product backlog is defined. (Schwaber & Beedle 2002).

3.2 Conceptual model of productization

Many researchers divide the product or service into a core or standardized part and a customer value adding extension part (Artto et al. 2008, Crawford 1997, Simula et al. 2008). Simula et al. (2008) describe a conceptual model of productization. The model illustrates productization as a set of linking activities between the new product development and marketing functions. In other words, productization is a predecessor for a successful product entry, consisting of tasks.
that help the company to continuously develop the product by maximizing customer benefits for the organization’s goals. The success of the productization depends on how well the company understands the market needs (Flamholz 1995, Simula et al. 2008). There are some difficulties for companies to understand what is really meant by the productization phase, achieve a consensus within it and understand the prioritization of tasks in that phase (Simula et al. 2008). Simula et al. (2008) introduce a productization model consisting of two elements, inbound productization and outbound productization. The model is shown in Figure 6.

![Fig. 6. Conceptual model of productization concept (based on Simula et al. 2008).](image)

In Simula et al.’s (2008) model, the inbound activities enable the quality of the product (ability to make). During the outbound activities (ability to sell), the extended part is created on top of the core product. Flamholtz (1995) shares the same view, stating that inbound activities harmonize and systemize the offering delivery process and its outcome inside an organization. The main focus of the inbound productization is the development of a prototype up to a mature core product which can be produced repeatedly with optimized costs (Simula et al. 2008).

Simula et al. (2008) further concretize their model and propose that firms put effort into creating a comprehensive, extended product on top of the core product. The use of this extended part allows companies to reserve effort for the customization of the software product and increased capability to respond to customer needs. However, there is a risk for companies in applying the core product and extension product approach. They may end up in a situation where
consensus on the extended part is not clearly achieved. Furthermore, this may lead to a situation where the final product lacks some key elements to match the customer needs. Outbound productization is seen as an ability to sell and a phase to improve visibility and make the product into a concrete offering. It aims to increase the value perceived by customers and it is the final stage of the productization: once completed, the product is ready for the market (Simula et al. 2008).

3.3 Productization through modularity

Modularity in product development has been used as an answer for customized products. Modularity means creating separate components which can be mixed and combined in many ways in order to create multiple products. Using modular product development increases the agility and strategic flexibility of the company (Hänninen et al. 2012). According to Sanderson and Uzumeri (1990), the following methods can be applied to achieve modularity in product development:

1. One or more basic modules are used for all products, and some additional modules can be added on top of the basic modules to create a final product.
2. The final product consists of modules which are put together without any so-called basic module(s).

The platform approach introduced in Section 2.3.4 represents one form of modular product development. Platforms establish one basic module, and multiple products or product families can be built on top of it. Modular product development is often associated with physical product development, but it can also be applied in service and solution development (Hänninen et al. 2012, Sanderson & Uzumeri 1990).

As a platform consists of common components, modules or parts which can be combined in several ways into a derivative products (Meyer & Lehnerd 2000, Sääksjärvi 2002), the productization through modularity has similarities with COTS (Commercial-Off-The-Shelf) and MOTS (Modified-Off-The-Shelf) software product development. COTS and MOTS software products are Off-The-Self (OTS) software products which can be considered as productized software products because the development is conducted outside the developers’ control (Carmel & Becker 1995, Mäntyniemi et al. 2004).
4 Interaction

This chapter introduces a theoretical view of interaction, in order to address the problem of stakeholder interactions in cross-functional software productization.

4.1 Introduction

Organizations consist of interrelated business units, departments, teams and working groups rather than merely individuals (Hogg et al. 2012). Organizational groups do not operate in isolation (Bourne & Walker 2005), and therefore organizations themselves need to join forces to collaborate towards common objectives (Hogg et al. 2012). Interactions and overlapping of development stages between different functions are considered important factors in trying to achieve shorter development lead times (Lin et al. 2009). The term ‘interaction’ can be defined as mutual influence of persons and things on each other (MWOB 2013). Liljander and Strandvik (1995) define interaction through a typology of ‘episodes’ and ‘relationships’, episodes being one-time exchanges of value between two parties with no prior or subsequent interaction, while relationships are continuous by nature, and associated with commitment, trust and loyalty. In summary, interaction can be referred to simply as two-way communication (Akhbar & Hassan 2010).

Interactions allow information exchange and knowledge generation, co-design, co-creation and consumption of services; in other words, interactions “constitute the very fabric of exchange” (Berthon & John 2006). The concept of interaction considers multiple and various groups of interactants such as leaders and managers and subordinates, employees and customers, employees and co-workers, customers and other customers (Eckert et al. 2001). Social interaction enhances cross-functional high-quality knowledge exchange and stimulates cooperation in achieving goals (Tsai & Ghoshal 1998).

4.2 Stakeholder interactions

The product development process is complex by nature (Safayeni et al. 2008), involving multiple functions (Brettel et al. 2011) consisting of a vast exchange of information across tasks. A project is seen as a multi-functional and collaborative task (Wang et al. 2002, Zhang 2012), and it is important to identify stakeholders and their significance through their interest and power relevant to the project.
However, it should be noted that their significance may be unstable and variable (Ackerman & Eden 2011). The unpredictable and frequently fast-changing environment demands flexibility and fast responsiveness, which is enabled by established feedback and reflection mechanisms feeding the experience into projects to come (Jaafari 2003).

Functional interactions and overlapping are essential enablers in accelerating the product development process and reducing project uncertainties (Joglekar & Ford 2005, Smith & Reinertsen 1998). Collaborative design is important in order to satisfy market needs and requirements through high innovation, high quality, reduced costs and faster time-to-market (Vijaykuar & Chakrabarti 2007). According to scholars, the importance of information and knowledge transfer is understood and recognized, but is very difficult to achieve (Luo et al. 2006).

A variety of terms with fairly vague definitions are used by scholars to describe stakeholders’ inter-related connections. Chen and Huang (2007) use terms such as social interpersonal interaction and interactive relationships. Bourne (2006) uses the term relationships in visualizing and assessing stakeholders’ influence and expectations. Ackerman and Eden (2011) use interaction, interconnection and interface; and Freeman and McVea (2001) use the term interconnections. This leads to an impression that these terms can be considered as synonyms of interaction in the context of stakeholders’ inter-related connections.

The term interaction is defined as “mutual or reciprocal action or influence” and the term interact as “to talk or do things with other people” (MWOB 2013). The term interaction is used in this study because mutual influence is highlighted in the definitions, and interactions are the focal point of the research problem. In this study, the concept of stakeholder interaction refers to a social interpersonal interaction between stakeholders. Team interaction is carried out within a team and can be characterized to be adaptive and maintain high levels of collective performance even if the team and circumstances are adverse (Zaccaro et al. 2001). Team interaction is not the focus of this study. In social interpersonal interaction, members of an organization interact with each other through communication, trust and coordination (Chen & Huang 2007). Mutual understanding and trust enables organizational knowledge integration and distribution (Bartol & Srivastava 2002). Trust between individuals increases willingness to exchange and absorb information (Tsai & Ghoshal 1998). According to Bourne (2006), effective communication is a key element in the process of establishing and maintaining relationships. The term relationship has similarities with interaction,
and these terms are synonymously by some scholars (Chen & Huang 2007, Bourne 2006).

The starting point for understanding stakeholder interactions is identifying the influential stakeholders for a project. Frooman (1999) asks very basic questions: “Who are the stakeholders?”, “What do they want?” and “How are they trying to get it?” These questions are useful for any project manager starting to identify the net of interactions relevant for the project. According to Olausson (2009), the use of both formal methods and quasi-formalized tools for rapid and regular interaction add the ability to handle uncertainty and complexity in fast-paced development projects. The strategies and activities should be integrated and synchronized among project team members.

The typical functions of product development are administration, research and development, sales and marketing, and operations. Large companies can also be structured as business units which are responsible for particular product categories or market areas. Although all objectives should be aligned and support the company goals and strategy, project managers face numerous challenges in trying to consider all of the needs and desires received from different functions, and this seems to be a very common management challenge. Product-related decisions are not driven only by profits and return on investments but also by customer demands and requirement, availability of technology, manufacturing and logistics capability, and safety and liability (Pesonen 2001). Chen (2007) stresses the importance of team interaction, stating that communication and coordination are critical factors for project performance.

While stakeholder theory has focused on organizational interactions with independent stakeholder relationships (Freeman 1984), recent research has raised the importance of the network of interactions and stakeholder multiplicity (Frooman 1999, Neville & Menguc 2006, Rowley 1997). Recognition of the roles of stakeholder groups in this network enhances the understanding of their influence on the organization (Neville & Menguc 2006, Wolfe & Putler 2002). The management of the interactions between an organization’s stakeholders is one of the most important tasks in the implementation of a company’s strategic goals (Ackerman & Eden 2011). The identification of stakeholders and the relationships between them is crucial (Freeman & McVea 2001). Stakeholders’ response to organizational activities may generate dynamic responses throughout other stakeholders (Ackerman & Eden 2011).

Capability of managing stakeholder relationships is a key factor in creating successful project deliverables and meeting stakeholders’ expectations (Bourne &
Stakeholder interactions have been considered in stakeholder theory (Rowley 1997), but there is a lack of detailed research (Neville & Menguc 2006). Stakeholder interactions are important in ensuring that the interests of different stakeholders are acknowledged and understood. Rowley (1997) widens stakeholder theory by arguing that stakeholder relationships occur in a network of influences. It is important to consider each stakeholder’s interests as individuals, but also collectively including cultural and power issues (Bourne & Walker 2005).

The interaction process between actors enables value creation for and with each other (Grönroos 2008, Payne et al. 2008). Freeman (1984) considers stakeholder management as a leadership issue and explains that the most beneficial way in which to interact with stakeholders lies in the leadership style and tone. Rowley (1997) argues that companies do not react to stakeholders individually but instead to the interaction of multiple influences from the entire stakeholder set. In order to explain stakeholders’ influence on a firm’s operations, the simultaneous interdependent interactions of stakeholders must be considered (Rowley 1997).

The relationships between stakeholders are dynamic and complex by nature, and they evolve over time and business. Stakeholder relationships arise between people, meaning that personal values and backgrounds influence the relationships (Talvio & Välimaa 2004). In a large and complex project, it is a challenging task to ensure that all relevant stakeholders’ interests are considered. According to stakeholder theory, no stakeholder has primacy over another, even when the project manager must balance conflicting demands (Greenwood & De Cieri 2005).

Stakeholders may have competing values, even though they need to cooperate in order to be responsive to market needs. Conflicts and tension are expected to arise in an organization during new product development when some of the stakeholders represent a strong market view. Organizations try to manage these conflicts by having several multiple and overlapping prioritization methods and coordinated interaction mechanisms (Driessen & Hillebrand 2010). Competition exists on many levels of the company, such as between business units, departments and groups (Tsai 2002). The degree of competition across functions is a critical factor that impedes efficient information transfer across boundaries within an organization, which leads to the question whether competing functions are able to cooperate to enhance the organizational performance. In practice, the functional areas within the company are compelled to compete and cooperate, but in turn cooperation and competition should be stimulated across functions to improve knowledge transfer within the company, leading to improved financial and customer performance (Luo et al. 2006).
Communication and coordination are needed to reach common agreement on the common goal and the content of the product. Managing project stakeholders above and below the project manager may have positive or detrimental impacts on large-scale projects. Since various stakeholders have or perceive themselves to have stakes regarding the project which may be congruent or incongruent with those of the project manager, it is essential to understand the objectives of each stakeholder. In addition, the needs and interests of stakeholders must be managed and adapted throughout the project lifecycle and project management process by the project manager (Sutterfield et al. 2006). For example, misunderstandings between marketing and customers or inability to transform customer requirements into product attributes can lead to configuration errors (Pringle 2001).

4.3 Elements of interaction

As stated earlier, product development is a joint effort including numerous interactions. Cross-functional software development embraces effective communication and concrete collaboration between engineers, product managers, and designers, among other (Botzenhard et al. 2011). Overlapping development stages and cross-functional interactions are key elements for shortening new product development lead time, even though functional interaction requires more communication and overlapping functions are more costly (Lin et al. 2009). Overlapping in this context refers to a product development process in which the downstream phase is started prior to the completion of the upstream phase. Dougherty (1992) describes functional units within organizations as “different thought worlds” which provide separate and divergent solutions to the common problem, resulting in defensive behavior.

Chen (2007) emphasizes the role of communication and coordination as enablers for successful cross-functional team interaction. Vijaykumar and Chakrabarti (2007) mention factors for successful collaboration such as communication, decision-making, social issues, and process management. They consider communication as a key element of collaboration and see it as being influenced by quality of interaction. Kahn (1996) sees cross-functional integration as consisting of two dimensions: interaction and collaboration. The success of interaction is seen in light of formal coordination mechanisms, and collaboration reflects informal harmonious collaborative interdepartmental relationships (Atuahene-Gima 2005, Ruekert & Walker 1987).
The reconciliation of different expectations and conflicts is called coordination, in which organizations must define the problem, identify the options available for the problem, and eventually create a common solution and collaboration across specialized functions (Oliva & Watson 2007). The literature of organizational behavior and operations management refers to this collaboration as integration (Barratt 2004, Ellinger 2000, Kahn 1996). Social interaction enhances cross-functional high quality knowledge exchange and stimulates cooperation in achieving goals (Tsai & Ghoshal 1998). Communication, collaboration and integration as elements of interaction are introduced next.

4.3.1 Communication

Communication is a major element in collaboration, and its quality is heavily influenced by the quality of interaction (Vijaykumar & Chakrabarti 2007). Functional interactions increase the communication time in the company (Lin et al. 2009). The term communication is defined as “imparting or interchanging thoughts, opinions, or information by speech, writing, or signs” (Mishra et al. 2012:1067). According to Eppinger (2001), even when the product development process itself is often described as and seems to be one-directional information transfer, the actual communication is bi-directional information exchange. Understanding of different stakeholders’ demands through their interest level and power impact on the project helps to build relationships and improve communication, and diminishes the danger of assumptions (Olander & Landin 2005). Communication has been seen as crucial for project managers for establishing relationships with not only tame and supportive stakeholders but also hostile stakeholders who argue with the priorities of the project goals (Bourne & Walker 2005). Maintaining active communication helps in observing early warnings of stakeholders’ conceptions (Briner et al. 1996).

As new product development is a knowledge-based activity, the communication is about sharing explicit and tacit knowledge. Explicit knowledge can be referred to as ‘know-that’ and as a synonym for information. It can be identified, explained, captured in systems and databases, and shared. Tacit knowledge is difficult to articulate and record, and is based on experience (Goffin et al. 2010). Tacit knowledge can also be called ‘know-how’ (Brown & Duguid 2001, Goffin et al. 2010), and can be considered as a process of learning (Wilson 2002). According to Nonaka (1994b), personal interaction enables tacit
knowledge sharing in a work context. Explicit and tacit knowledge are characterized in Table 6.

### Table 6. Explicit and tacit knowledge (based on Goffin et al. 2010).

<table>
<thead>
<tr>
<th>Nature</th>
<th>Explicit knowledge</th>
<th>Tacit knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easily identifiable</td>
<td>Within person knowledge</td>
<td></td>
</tr>
<tr>
<td>Relatively easy to share</td>
<td>Difficult to articulate</td>
<td></td>
</tr>
<tr>
<td>Intrinsically incomplete; lacks context and requires interpretation</td>
<td>Hard to share</td>
<td>Can be shared only indirectly</td>
</tr>
<tr>
<td>Typical examples</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information</td>
<td>Intuition and insight</td>
<td></td>
</tr>
<tr>
<td>Know-that</td>
<td>Practical intelligence, skills, and practice</td>
<td></td>
</tr>
<tr>
<td>Theoretical knowledge</td>
<td>Know-how and heuristics</td>
<td></td>
</tr>
<tr>
<td>Rules of thumb</td>
<td>Rules of thumb</td>
<td></td>
</tr>
<tr>
<td>Mental models and beliefs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mechanisms for generating and sharing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Codification</td>
<td>Practice</td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>Personal and team reflection</td>
<td></td>
</tr>
<tr>
<td>Databases and search engines</td>
<td>Drawing mental maps</td>
<td></td>
</tr>
<tr>
<td>Blogs, wikis, and intranets</td>
<td>Apprenticeships</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Social interaction and mentoring</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Story-telling and metaphors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>New codification systems can make some tacit knowledge easier to share, through converting some elements of it into explicit knowledge</td>
<td></td>
</tr>
<tr>
<td>Key issues in new product development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage the creation, storage, and retrieval of explicit knowledge</td>
<td>Recognize people as source of tacit knowledge</td>
<td></td>
</tr>
<tr>
<td>Motivate R&amp;D personnel to produce thorough documentation</td>
<td>Create space for networks, informal interactions, and trust</td>
<td></td>
</tr>
<tr>
<td>Capture explicit knowledge generated in post-project-reviews</td>
<td>Stimulate knowledge flow between teams</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use tacit knowledge for competitive advantage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Encourage knowledge sharing so that knowledge is not lost</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Facilitate post-project-reviews to support the generation of tacit knowledge</td>
<td></td>
</tr>
</tbody>
</table>

Communication happens in various ways on many different levels across functions. Bourne & Walker (2005) emphasizes the need for communication not only with positive stakeholders but also with those who are a threat to project
goals. Stakeholders can be engaged into active involvement with projects if project managers establish an open and transparent mechanism for communication and information sharing. Stakeholders interact with each other by exchanging information and knowledge, and this is seen a major strategic means for addressing problems (Nilson & Fageström 2006). Jassawalla and Sashittal (1998) report a strong association between effective cross-functional collaboration and transparency – a condition of high awareness achieved as a result of intense communication. Also, according to Cooper (1996) high-quality teams interact and communicate well and often. Cross-functional new product development teams must obtain information and resources from diverse sources inside and outside the organization. Key to this is the team’s skill at boundary management – the process by which teams initiate interactions with, and respond to communications from, other parts of the organization, both vertically and laterally (Holland et al. 2000).

A lack of appropriate information flow in large-scale engineering processes may lead to a number of problems among designers and decision-makers (Eckert et al. 2001). Problems of inadequate communication flow include not understanding the big picture, not knowing what to do, different interpretations, and information distortion (Vijaykumar & Chakrabarti 2007). Communication problems are described as common in cross-functional integration (Kusunoki & Numagami 1998). Kleinsmann and Valkenburg (2003) acknowledge three levels of barriers to communication in collaborative design: participants, project and organizational barriers. People prefer to communicate most frequently within their own teams and less with other functions or teams Curtis et al. (1988). Organizations are structured very commonly by functions with their own objectives and special language, leading to cross-communication and coordination difficulties (Safayeni et al. 2008). Insufficient and deficient communication is likely to cause disagreements and tension in a cross-functional team in the sense that, due to extensive disagreements, team members tend to commit to their individual targets and tasks, and common understanding is more difficult to achieve (Chen 2007, Dougherty 1992).

4.3.2 Collaboration

Collaboration as a term originates in management literature, referring to a method in which competing interests may lead to win-win results. Roschelle and Teasley (1995:70) define collaboration as “coordinated, synchronous activity that is the
result of a continued attempt to construct and maintain a shared conception of a problem”. In the new product development context, collaboration is often used interchangeably with cooperation between functional teams (Jassawalla & Sashittal 1999). Collaboration between technical, marketing, manufacturing and sales departments leads to successful products (Dougherty 1992). Kahn (1996) studied the impacts of collaboration and interaction for product development and product management (post-launch) performance and found that collaboration has a positive and strong effect on performance. Interactions themselves did not seem to have a significant impact on product development performance, but the results indicated negative effects for exchange of documented information and meetings. Coordination of collaboration is the key success factor for project outcomes (Mat and Jantan 2009). Organizational culture may be an impediment to collaboration in a case where value and reward structures emphasize and support individual achievements (Orlikowski 1993). Uzzi (1997) presents the term ‘socially embedded’ as enhancing the ability to solve problems jointly by enabling actors to get feedback from each other, look for common solutions and integrate these solutions into daily operations. When collaborative actors confront different views, their mutual adjustment and engagement in problem-solving will improve and deepen (Heide & Miner 1992). Chen (2007) suggests that the more decentralized and less formal the organizational structure, the more favorable is cross-functional team interaction, and cross-functional team interaction correlates positively to new product development efficiency.

Group-specific aspirations and targets may invite conflicts between groups, causing obstacles for good cooperation and spilling over further. Organizational performance and efficiency are dependent on the capability to manage relationships between different groups (Knippenberg 2002). Functional units are interdependent, because each function has distinct competences, resources and capabilities (Ruekert & Walker 1987). According to Brett & Rognes (1986), there is a high potential for conflicts when groups have interdependencies and must rely on the expertise and results of other groups. Bailetti et al. (1998) and Zhang (2012) recommend that the coordination of the development of complex products should rely on the interdependence of responsibilities in functionally oriented companies. Although groups are established to work rationally on a task, powerful unconscious forces exist within them and influence their behavior. Holland et al. (2000) identify some key issues causing tension between team goals and functional priorities: conflicting organizational goals, competition for resources, overlapping responsibilities, conflicting personal goals, no clear direction or
priorities and lack of cooperation. Intragroup collaboration particularly meets obstacles if groups are central and perceive themselves to be in a competitive situation, and feel that their identity as a group is threatened, even though in reality this is only a perception (Hornsey & Hogg 2000).

Collaborative working is difficult to learn and is not a skill that can be assumed based on membership of a team. Some people adopt collaborative working behaviors and are able to accelerate and contribute to common processes. Some people are challenged by personal interactions and commitments to common targets. If collaboration is poor, it may elevate cynicism and weak morale among team members (Jassawalla & Sashittal 1999). Intergroup behavior is driven by the subjective perceptions and experiences of individuals (Hogg et al. 2012). The literature shows that inter-functional conflict and inter-functional connectedness are key factors in interactions between functional units (Cadogan et al. 2005). Employee satisfaction is an important enabler for successful interaction and collaboration because it increases employees’ commitment level and decreases inter-functional conflicts and tensions, making the integration of inter-functional goals more certain - during conflicts and tension, contact with other functional groups is likely to be avoided (Menon et al. 1997, Cadogan et al. 2005). Cross-functional competition may lead to comparisons between functional units (Maltz & Kohli 1996), and to different goals and priorities (Houston et al. 2001, Ruekert & Walker 1987). Cooperation between competing units can absorb know-how across units through frequent interactions. Productive interactions can be stimulated when units compete for resources because they are likely more willing to share and exploit information and ideas with each other (Tsai 2002). Luo et al. (2006) present the concept of ‘coopetition’, defining it as simultaneous cooperation and competition within the company. The theory of coopetition is founded in sociological research and literature. In cross-functional coopetition, inter-organizational interactions can be both cooperative and competitive by nature. Cooperation comes about through interdepartmental collaboration through knowledge transfer and sharing in the common interest of the company. The simultaneous occurrence of cross-functional cooperation and competition improves company performance through learning, cost savings, innovation, and resources (Luo et al. 2006).

Leadership of intergroup interactions in situations like is very challenging. Intergroup competition may cause tensions, for example competing for resources and differences in objectives for different groups, which leads groups to drive their own interests and prevent high-quality intergroup collaboration and
functioning (Hogg et al. 2012). Lovelace et al. (2001) argue further that close collaborative relationships between functional managers should be encouraged. Managing the interfaces between R&D and other functions is difficult because of diversity in roles, responsibilities and skills which easily lead to conflicting priorities and needs (Webber 2002, Xie et al. 2003). Conflicts and inter-departmental barriers can be reduced by setting clear common goals (Xie et al. 2003), and organizing multi-functional training activities (Maltz & Kohli 2000) and joint reward systems (Wei & Atuahene-Gima 2009).

4.3.3 Integration

A company needs to ensure buy-in by all of its functional areas through cross-functional integration (De Luca & Atuahene-Gima 2007, Lovelace et al. 2001), to be able to develop new products and services (Srivastava et al. 1998). Internal integration is explained as cross-functional team orientation linking organizational teams and functions horizontally (Bishop 1999, Henke et al. 1993, Teixeira et al. 2012). Well-adopted cross-functional integration results in rich collaboration and communication in the company, with increased mutual feedback and improved problem-solving ability (Olson et al. 2001, Teixeira 2012). In new product development literature, the term ‘integration’ is used as a joint term for different cross-functional linkages, and is often used interchangeably with ‘collaboration’. Both terms refer to different interests and people that want to achieve a common goal through interactions, information exchange, and coordination (Jassawalla & Sashittal 1998, Song et al. 1997, Souder & Moenaert 1992). The development of a new product requires cooperation and cross-functional integration (Moenaert et al. 1994). Interdepartmental integration consists of interaction and collaboration (Kahn 1996). Much of the literature assumes that effective integration corresponds with high levels of interaction (Griffin & Hauser 1996, Gupta & Wilemon 1986). Integration of expertise is crucial in knowledge organizations. The success of a company does not depend only on individuals and their competences but on how well their expertise is combined (Boh et al. 2007).

Cross-functional integration has been considered for long time to be a key success factor for successful products (Ernst 2002, Griffin and Hauser 1992, Gupta et al. 1986, 1996, Olson et al. 1995). Market requirements, technology choices, and product-related decisions are evaluated and traded off prior to actual product development phases, leading to formal or informal sets of requirements. 
and needs, technology choices, and project objectives (Tatikonda & Rosenthal 2000). Two forms of integration can be identified in cross-functional integration: internal and external. Internal integration consists of the project team members and project manager coordinating the project activities effectively and communicating the product concept. External integration has strong interaction with marketers, customers, and suppliers and aims to satisfy their needs (McNally et al. 2011).

Kahn (1996) describes inter-departmental integration as consisting of two elements: interaction and collaboration. Kahn and McDonough (1997) argue that there is strong evidence that interaction does not have significant impact on company performance and that documented information and meetings influence negatively. Griffin and Hauser (1996) and Gupta et al. (1986) claim that effective integration is based on high levels of interaction. Cross-functional development groups integrate different knowledge domains through individual experts which are needed for a successful and productive outcome (Souder 1981, 1988). Integration of cross-functional product development increases the number of meetings required for decision-making and communication (Song et al. 1998). Cross-functional integration is loaded with challenges, because people move from one functional area to another, bringing with them their different views (Griffin & Hauser 1996), cultures, and attitudes (Gupta & Wilemon 1986), making it difficult to integrate their knowledge in developing new products (De Luca & Atuahene-Gima 2007).

Different coordination mechanisms are more effective in some situation than others because interactions may differ among functional departments due to different circumstances and situational factors, for example if the environment is complex and very turbulent, or market/product development is very aggressive by nature, causing many dependencies for information and expertise between organizations and working groups (Ruekert & Walker 1987). When a product concept is unfamiliar and very new to a firm, the development process for the finished product is time-consuming and extensive (Ainamo 2007), but by using the right selection of techniques (Hardagon & Douglas 2001) and attitude (Boland & Collopy 2004), the process may be sped up. According to studies in organizational behavior, interdepartmental collaboration and common objectives and goals depend foremost on high-quality exchange relationships (Konovski & Pugh 1994).
5 Reference framework

This chapter describes the theoretical framework of the study used to guide the data collection and analysis. The concepts of previous chapters are combined theoretically and their relationships to each other are depicted. The emphasis of the reference framework will be placed on identifying the stakeholder interactions during cross-functional software productization in product development. The research framework consists of stakeholders, stakeholder key functions, and interactions. The framework is completed by the market view, which is the major source of new mobile product development needs and opportunities. The theoretical elements are presented in this chapter as structured in the a priori model.

5.1 The a priori model

The mobile software development context provides a basis for a field-directed a priori assumption on identification of stakeholder interactions and their capacity to influence the content of the mobile software product during the productization phase. Corporate management communicates the vision, mission and strategy to the organization. The company targets and product requirements are drawn from the strategy, or they may be communicated directly to product development by the corporate management. The complexity of product management is caused by interactions of stakeholders. Achieving a common understanding about the product to be developed is very challenging because of the numerous company stakeholder needs, shared strategic ones, competing expectations, and also some personal aspirations (Weerd et al. 2006a).

From a theoretical approach, the a priori model presented in Figure 7 consists of mobile market needs and opportunities, stakeholders and interactions. Stakeholders and their key functions are explained in Section 2.4. The interactions are described through communication, collaboration and integration, as presented in Chapter 4.
Stakeholders and their key functions represent the main responsibilities regarding product development. Stakeholders are considered in the a priori model according to Freeman’s (1984:46) definition: “A stakeholder in an organization is any group or individual who can affect or be affected by the achievement of the organization’s objectives.” Communication, collaboration and integration reflect the form of interactions and major outcomes carried out. Interaction is defined as: “mutual or reciprocal action or influence” according to MWOB (2013). Communication is seen as “impacting or interchanging thoughts, opinions, or information by speech, writing, or signs” as stated by Mishra et al. (2012:1067). Collaboration is seen as interchangeable with cooperation consisting of knowledge transfer and sharing (Jassawalla & Sashittal 1999) and as “coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem” (Roschelle & Teasley 1995:70). The mobile market is a source of information to stakeholders. The process of listening to market and customers is iterative, aiming to optimize the implementation of valued delivery systems to the market through the feedback loop. A market opportunity is referred to, according to Bond and Houston (2003), as the
intersection of the company capability and customer need in which new technologies or emerging customer needs may create new market opportunities. (Bond & Houston 2003). The fundamental idea of productization is to build something to become a commercial product (Flamholtz & Hua 2002), and it is applied in the a priori model too. Productization can be described as a process or series of steps starting from product or service planning through development/manufacturing and ending with marketing activities (Flamholtz & Aksehirli 2000, Hietala et al. 2004). In the software product business, productization is explained as being “the process of transforming from customer specific software development to a standard software product.” (Artz et al. 2010:7).

Stakeholders and functions in the a priori model are organized sequentially to make the analysis more systematic. It should be noted that the order of the functions of stakeholders is not taken into account in the a priori model. The aim of the illustration is to emphasize the major functions and responsibility areas of each stakeholder group. The arrows directing to stakeholders indicate feedback from the market which may be received by any stakeholder at any time to be iteratively processed and being incorporated into software productization. The stakeholders and their functions are based on Weerd’s et al. (2006) software product management framework presented in Section 2.4.3. The a priori model is structured in the form of stakeholder-based swim lanes to show the stakeholders and their key functions. The model presents the matrix of stakeholders and the type of their interactions based on the theory. The a priori model is viewed through two dimensions. One dimension of the model is the stakeholders’ view, indicated by seven types of stakeholder and their key functions in software productization. The stakeholders and their functions are explained in more detail in the following sections. The second dimension is the interaction view, consisting of three elements: communication, collaboration and integration. Communication focuses on how stakeholders communicate with software productization. Collaboration concentrates on the forms of cooperation between stakeholders and software productization. Integration is concerned with stakeholders and how they are integrated with software productization. Product development delivers the final products to the mobile market, which is also the source of market needs and opportunities.

The a priori model was developed from the theoretical literature to convey a more understandable schema of this study. This research aims to contribute to the literature on stakeholder interactions in software productization. The a priori
model is used as a lens with which to analyze the research questions. The elements of the a priori model are introduced next.

5.1.1 Mobile market needs and opportunities

The phenomenon of mobile products has amazed the world. People have adopted the use of mobile products on a large scale in less than two decades. The technology, performance, and miniaturization of mobile products have fueled the enormous growth in the number of mobile users. Inexpensive products and the reasonable cost of connections have led to the growth of new consumer groups such as those in the emerging economies. Consumer-based market segmentation is more and more prevalent, reaching more and more satisfied users.

Market information is an immense source of mobile opportunities (Heurese et al. 2012). It consists of customer’s needs (Fein et al. 2011) and information about competitors (Weed et al. 2006a). Marketing is in a key role in gathering and analyzing the market data (Fein et al. 2011), and it is essential that communication between marketing and new product development is ensured in order to create successful products which respond to customer and market needs (Dougherty 1992, Griffin & Hauser 1996). Market needs and opportunities in the a priori model are considered as an abstract stakeholder from the product development point of view, corroborating potential customers and competitors.

5.1.2 Stakeholders

Stakeholders are vital for the company’s survival and success (Freeman 2002). Freeman’s (1984:46) defined a term stakeholder as follows: “A stakeholder in an organization is any group or individual who can affect or be affected by the achievement of the organization’s objectives”, and this is applied in this research, too. The a priori model is focused on company internal stakeholders, who constitute and influence daily operations (Mark-Herbert & Schanz 2007) such as product development, and related stakeholders. From a product development perspective, there are numerous stakeholders whose demands and expectations should be addressed (Bourne & Walker 2005). Sometimes expectations are driven by personal ambitions or organizational objective setting and stakeholders are eager to get their own views to the top of the priority list (Fassin 2008). Their needs and demands may be conflicting or even competing, and this makes the situation very challenging for the product development manager (Driessen &
Stakeholders express their needs in many ways. However, stakeholder groups have different weights, and as such their expectations cannot be treated equally (Fassin 2008, Harrison & Freeman 1999). Stakeholders and stakeholder groups may have different roles within a group (Fassin 2008). Although stakeholders are heterogeneous in a product development context they share the same interests (Post et al. 2002). The most influential ones, having the most power over decisions regarding resourcing and funding, should be acknowledged. In order to understand their power, stakeholders should be first identified, categorized and analyzed. Their needs should be prioritized, because an inability to satisfy these expectations may lead to product failure and poor outcomes (Bourne & Walker 2005, Rowley 1997). Stakeholders’ functions are considered as responsibilities, activities and tasks carried out by the stakeholders. Stakeholder key functions were introduced and highlighted in italics in Section 2.3 and are summarized in Table 7.
Table 7. Key functions of stakeholders as stated in Section 2.3.

<table>
<thead>
<tr>
<th>Stakeholders</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Gathering, sharing and using information about the market</td>
</tr>
<tr>
<td></td>
<td>Gathering new product requirements</td>
</tr>
<tr>
<td></td>
<td>Disseminating market information</td>
</tr>
<tr>
<td></td>
<td>Channeling new product requirements to be processed further in the company</td>
</tr>
<tr>
<td></td>
<td>Setting new product objectives and product features, and resolving cost-design-performance trade-offs</td>
</tr>
<tr>
<td></td>
<td>Linking needs and requirements outside the company</td>
</tr>
<tr>
<td></td>
<td>Spanning the functional boundaries in the company</td>
</tr>
<tr>
<td></td>
<td>Defining product strategy and product roadmap, tearing the roadmap down to business requirements</td>
</tr>
<tr>
<td></td>
<td>Representing and being the voice of the customer</td>
</tr>
<tr>
<td></td>
<td>Planning go-to-market and ensuring product support</td>
</tr>
<tr>
<td>Corporate management</td>
<td>Having discretion over decisions in the company</td>
</tr>
<tr>
<td></td>
<td>Gathering, sharing, and using information about the market</td>
</tr>
<tr>
<td></td>
<td>Creating a strategy map</td>
</tr>
<tr>
<td></td>
<td>Having overall responsibility for the product strategy</td>
</tr>
<tr>
<td></td>
<td>Communicating strategy, vision, and mission</td>
</tr>
<tr>
<td></td>
<td>Supervising functions</td>
</tr>
<tr>
<td></td>
<td>Sponsoring projects</td>
</tr>
<tr>
<td></td>
<td>Maintaining commitment</td>
</tr>
<tr>
<td></td>
<td>Defining the vision of the product</td>
</tr>
<tr>
<td>Portfolio management</td>
<td>Long-term planning of new product releases</td>
</tr>
<tr>
<td></td>
<td>Mapping new product releases for market windows</td>
</tr>
<tr>
<td></td>
<td>Bridging requirements for future software releases</td>
</tr>
<tr>
<td></td>
<td>Combining product positions and development aspects</td>
</tr>
<tr>
<td></td>
<td>Connecting management, marketing, and product development functions</td>
</tr>
<tr>
<td></td>
<td>Linking business and customer needs</td>
</tr>
<tr>
<td></td>
<td>High-level content definition for future products</td>
</tr>
<tr>
<td></td>
<td>Looking for new business and product opportunities from market pull and technology push aspects</td>
</tr>
<tr>
<td></td>
<td>Providing a strategic path and visibility of products via key themes of products and core assets</td>
</tr>
<tr>
<td>Roadmapping</td>
<td>Long-term planning of new product releases and upgrades, mapping those for identified or assumed market windows</td>
</tr>
<tr>
<td></td>
<td>Bridging the gap between business and product development</td>
</tr>
<tr>
<td></td>
<td>Combining both product positions and development aspects as well as connecting the management, marketing, and product development functions to build and manage the long-term high-level development plan linking business and customer needs</td>
</tr>
<tr>
<td></td>
<td>High-level content definition for future product</td>
</tr>
<tr>
<td></td>
<td>Looking for new business and product opportunities from market pull and technology push</td>
</tr>
</tbody>
</table>
Marketing is responsible for gathering, processing and delivering the market information further to other stakeholders. All other stakeholders need market information in order to make appropriate decisions in their own responsibility area. Corporate management is responsible for sharing and communicating the company strategy, vision and mission to all employees in the company. It has the overall decision-making authority and uses its power in strategic decision-making. Corporate management is also responsible for enabling the environment, resources and funding for company operations. Market needs and opportunities are fed into development units through the portfolio management, product roadmapping, and release and requirement management processes. Portfolio management is responsible for creating and maintaining a long-term product and service map consisting of evaluating, prioritizing, and selecting new projects and existing ones. It balances the company’s development projects and resource management (Archer & Ghasemzadeh 1999). Roadmapping creates long-term product release planning by positioning products against identified market opportunities (Artz et al. 2010) and combining business targets, customer needs, and product development (Lehtola et al. 2005, Vähäniitty et al. 2002). Because of the many different stakeholder needs and aspirations, it is difficult to decide on
the most important requirements for software releases. Release planning and requirement management is responsible for planning and prioritization software releases and their content. Platform development provides the basic technological capability for mobile product development. Platform development has been seen as an answer for accelerating product development. Finally, product development projects create value delivery systems for the market. They are established to fulfill the company’s business targets.

5.1.3 Cross-functional software productization

Product development consists of set of activities to be performed to build and finalize the commercial product and its entry into the market segment (Artz et al. 2010, Hänninen et al. 2012, Kotler 2003, Lehtimäki et al. 2008). The productization phase consists of the integration of all relevant components into a final product Brettel et al. 2011, Olson et al. 2001). All products have some features and properties that can be considered as a core of that product. In productization, a set of differentiated features and functionalities are added on top of the core product through a single product project that makes the tailoring work for the product (Simula et al. 2008). This view is also applied in the a priori model.

Mobile software development is a complex and cross-functional task involving numerous stakeholders, hence a project manager faces daily challenges dealing with a complex network of interactions (Clarkson 1991, Lim et al. 2005). Functions represent different capabilities of stakeholders (Fain et al. 2011). Software product development is guided by a disciplined process called software product management, which should combine the business and engineering perspectives (Weerd et al. 2006a). Grönroos (2008) and Payne et al. (2008) state that the interaction between actors promotes value creation for and with each other. The common goal is the guiding element for cross-functional product development. Cross-functional knowledge exchange stimulates effective cross-functional cooperation (Tsai & Ghoshal 1998).

5.1.4 Interactions

Interaction-related issues are the problem domain of this research. The interactions are examined through the mobile software productization perspective. The project manager is the key player trying to balance the stakeholder needs,
product development cost and schedule, and technological capabilities. It is common for a project manager to face challenges in trying to consider and balance all of the demands of the stakeholders. In order to understand the interactions, the stakeholders must be identified (Freeman & McVea 2001). It is crucial for a product development project to get the all needed information and knowledge about the market and customer requirements and needs in order to develop a successful product (Bourne & Walker 2005). Conflicts and competing requirements occur between stakeholders, functions and individuals due to organizational boundaries, but stakeholders’ individual ambitions and aspirations can also make product development very difficult (Greenwood & Cieri 2005, Nilson & Fagerström 2006). To ensure a successful outcome, it is essential for the project manager to identify the most important stakeholders and their expectations, and analyze their influence. It should be noted that not all stakeholders are or should be treated as equal, because their power and intensity of interactions vary (Phillips 2003, Post et al. 2002).

There are interactions between software productization and stakeholders, but it must be noted that stakeholders also interact with each other, and this web of interactions has an impact on product development. Stakeholder interactions are dynamic by nature and evolve over time (Talvio & Välimaa 2004). The term interaction is defined through typology of ‘episodes’ and ‘relationships’, with episodes being one-time exchanges of value between two parties with no prior or subsequent interaction, while relationships are continuous by nature, and associated with commitment, trust and loyalty (Liljander & Strandvik (1995). Interactions can be seen simply as mutual influence of persons (MWOB 2013) and information exchange and knowledge generation (Berthon & Joby 2006). Communication and coordination are crucial factors for project performance (Chen 2007). The types of interactions of the a priori model are explained in the next section.

**Communication**

Communication is a central element of product development. Efficient and effective communication is often embedded in formal processes in the form of regular meetings, memos and status reports to ensure the appropriate access to information for all relevant parties and stakeholders. Lack of communication may be perceived as intentional information hiding, which increases mistrust between different stakeholders. If the overall objectives are not communicated clearly, the
common target of the company will remain fuzzy and people will not be able to work efficiently towards common goals (Chen 2007, Dougherty 1992).

Stakeholders become engaged with product development through communication. Communication is needed with both positive and negative stakeholders (Bourne & Walker 2005), in order to decrease assumptions and false beliefs (Olander & Landin 2005) and address and solve problems (Nilson & Fageström 2006). Communication is bi-directional information exchange (Eppinger 2001). It happens in various ways through different functions. The term communication is understood in the a priori model as “imparting or interchanging thoughts, opinions, or information by speech, writing, or signs”, as defined by Mishra et al. (2012:1067). Poor and inadequate information flow in large-scale engineering processes may lead to problems among project development and decision-makers (Eckert et al. 2001).

**Collaboration**

Collaboration has a positive impact on a company’s performance and is a major element in project outcomes (Dougherty 1992, Kahn 1996). Cross-functional product development has generated the need for mechanisms of collaboration. The ability to collaborate is not always an obvious interpersonal skill, although it is expected in the cross-functional environment. Some people may have challenges in learning and adopting collaborative working methods, and interpersonal issues may hinder collaboration. Poorly implemented collaboration between cross-functional teams may increase cynicism and suspicious attitudes among team members (Jassawalla & Sashittal 1999). Cross-functional groups may have different aspirations and targets, so cross-functional collaboration is a good territory for conflict (Knippenberg 2002). Transparency is one way to improve collaboration. Transparency refers to a high level of clarity achieved by collaborating teams through communication and intensive information exchange (Jassawalla & Sashittal 1999).

**Integration**

Integration ensures that all of a company’s functional areas undertake and commit themselves to developing new products and services (Srivastava et al. 1998). Integration is carried out through interactions (Ylääranta 2006). Integration enables different knowledge domains of the company, creates a basis for optimal
decision-making and is considered to be one of the key factors for successful project outcomes (Jassawalla & Sashittal 1998, Souder 1981, 1988).

According to Olson et al. (2001) and Teixeira (2012), cross-functional integration establishes collaboration and communication, and increases mutual feedback and increased problem-solving ability. Internal integration improves collaboration, communication, and problem-solving among departments (Olson et al. 2001, Teixeira et al. 2012).

5.2 Summary

The elements of the a priori model have been introduced. Stakeholders interact with each other and software productization during product development. Stakeholders perform functions in their operations. Product development projects need to exchange and use information and knowledge from different organizational stakeholders, which are considered as the product development stakeholders. A project manager is a hub which tries to balance all stakeholder needs. Software products are managed through a software management process consisting of portfolio management, roadmapping, and release planning and requirement management. The purpose of this process is to break down market needs and opportunities into clear requirements and smaller tasks for product development. Software project management is responsible for defining the features and functionalities of new software releases. The research method, case study and data collection are described in the next chapter.
6 Research design

One of the major goals of elaborating research design is to argue that ad hoc methods were not used, but the research was carried out in a rigorous way. Stake (1995) points out that good research does not only involve good methods used, but also good thinking, meaning that if a researcher does not understand what they are doing, methods may be of very little use. This chapter describes the philosophical basis of the research approach, the research strategy, and the empirical techniques applied. A starting point of any research and the methodology selection is to choose the ontological and epistemological paradigm.

6.1 Ontology, epistemology and methodology

All research relies on philosophical assumptions justifying the validity and the appropriate methodology chosen for the knowledge creation in a given study. In order to understand the research, it is important that these underpinning assumptions are explained to the reader. Before any research is done, it is necessary for the researcher to explore the most fundamental perceptions regarding the world in which they live. A continuous philosophical debate is ongoing, some approaches and disciplines to research become more appreciated at one time than at others, and major trends are extended. The approach adopted in this research is the qualitative case study. The philosophical approach of this study comes from the interpretive tradition. Researchers reflect their worldview of science through ontological, epistemological and methodological premises related to research paradigms and consistent research methods (Guba 1990).

It is important to position the paradigm of the research. A paradigm is considered by Guba and Lincoln (1994:107) as a “set of basic beliefs (or metaphysics) that deals with ultimate or first principles” representing a worldview that defines, for its holder, the nature of the world, the individual’s place in it, and the range of possible relationships to that world and its parts, as for example, cosmologies and theologies do”. Three fundamental questions are used to define the paradigms according to Guba and Lincoln (1994):

1. **Ontology.** What is the form and nature of reality and, therefore what is there that can be known about it?
2. **Epistemology.** What is the relationship between the knower of would-be knower and what can be known?
3. Methodology. How can the inquirer go about finding whatever he or she believes can be known?

Ontological and epistemological views concern what is commonly understood as a person’s worldview which affects their understanding of the reality and the question of knowledge creation.

**Ontology**: The term ontology is defined by Collins English Dictionary (2006) as “the branch of philosophy that deals with the nature of existence”. Wand and Weber (1993:220) define ontology as “a branch of philosophy concerned with articulating the nature and structure of the world”. Guba and Lincoln (1994) define ontological paradigms as having the following assumptions: positivism, post-positivism, critical theory, constructivism and realism. *Positivism* considers reality as real and apprehensible, often taking a cause-and-effect form. Reality does not depend on the observer’s perceptions. Positivism is regarded as the ‘received view’, and the goal is to quantify and control experiments. Knowledge is generated by statistical methods. In *post-positivism*, reality is seen as subjected to the widest critical examination to facilitate apprehending reality as closely as possible. The *critical theory* assumption is characterized by individual perceptions which are assessed by their feasibility to subjective aspects. In critical theory, virtual reality is shaped by social, economic, ethnic, political, cultural, and gender values, crystallized over time. In *constructivism*, reality is formed through individual views and the world consists of multiple constructed realities; these views cannot be compared with other individual perceptions. Constructivism sees reality as consisting of multiple local and specifically constructed intangible realities. In *realism*, reality is seen as being imperfectly real and probabilistically apprehensible, so triangulation from many sources is required to try to know it. Realism generates knowledge by extending findings through generalization and nests the empirical data within the theories.

**Epistemology** is defined by Hirschheim *et al.* (1995:20) as “the nature of human knowledge and understanding that can possibly be acquired through different types of inquiry and alternative methods of investigation”. Guba and Lincoln (1994) argue that epistemologically in *positivism* the researcher is able to study a phenomenon without having an impact on the actual phenomenon. In the positivism paradigm, findings are considered true, and a researcher is objective by viewing reality through one-way mirror. In *post-positivism*, replicated findings are probably true but always subject to falsification. Objectivity remains ideal but can be approximated and the importance of the external reviewer is emphasized. The
Constructivist approach presents the findings as created – a researcher is seen as a ‘passionate participant’ within the world being investigated. Findings are constructed through the process in which a researcher and the objective under study are merged into one. Critical theory sees findings mediated by values and a researcher is considered as a ‘transformative intellectual’ who changes the social world within which the participants live. The values of a researcher influence the research and the findings are created through interaction with the environment. The realistic scientific approach assumes that findings are probably true, and a researcher is value-aware and triangulates any perception they collect (Guba & Lincoln 1994).

Methodology consists of the techniques used during research to discover the truth. In the positivism paradigm, the methodology is seen as experimental and manipulative. Research questions and/or hypotheses are empirically tested and verified. Post-positivism favors methods that are used in natural settings (Guba & Lincoln 1994). The methodology emphasizes “critical multiplicity as a way of falsifying rather than verifying hypotheses” (Guba & Lincoln 1994:110).

The research at hand is an interpretive study, because it tries to understand the phenomena in the context in which the actors are assigned (Orlikowski & Baroudi 2002). Ontologically, this study is a post-positivist. This research is not positivist, because different participants might experience phenomena in the same context differently, nor critical, because the purpose is not to make changes to the operational mode in the case company during this research. The research method selected is a case study, because this is seen as a suitable method for understanding the behavior of people in a real-world context (Yin 2009). Epistemologically, this research is transactional and subjectivist. The researcher and the object under study are interlinked, and the findings are generated, ‘literally created’, alongside the research. The methodological approach assumes that social constructions are created through interactions between and among the researcher and the respondents (Guba & Lincoln 1994).

6.2 Qualitative case research

This research is a qualitative case study. Stake (1995) describes the differences between qualitative and quantitative studies through the elements of explanation and understanding as the purpose of the inquiry, the role of the researcher, and the knowledge discovered and constructed. The most distinguishing feature between qualitative and quantitative research is the knowledge that is searched for (Stake
Qualitative research is described as inductive rather than deductive, and it tends to be interpretive, seeking to understand the phenomenon in its context. It considers experiences rather than controls for variables as in an experiment. Qualitative research does not require a hypothesis in the beginning of the study as quantitative research always does (Guba & Lincoln 1994).

A qualitative research approach relying on the interpretive paradigm builds trustworthiness on credibility, transferability, dependability, and conformability (Denzin & Lincoln 1994). The credibility of this study is based on triangulation (member checking and negative case analysis). In terms of triangulation, interviews (received data from informants), participative observation, and document reviews (official and non-official documents and artifacts) are used. Triangulation is a mean of corroboration so that the researcher is confident with the findings and conclusions. Member checking was conducted in meetings and via telephone discussion to check the accuracy of my observations and artifacts. The findings were discussed, and members were asked to comment and feedback on my interpretations in several general meetings with the research focus organization members, informants and management team members, and also from those who did not participate as interviewed respondents. Member checking took place during the whole two years of data collection in numerous team, group and individual discussions and meetings. Negative case analysis was used to ensure rigor and in verification of the findings (Strauss & Corbin 1990). Negative case analysis was done by re-examining the case after the initial data analysis.

Qualitative research can be positivist, critical, or interpretive (Myers & Avison 2002, Orlikowski & Baroudi 2002). **Positivist** studies are used to test theory, and this approach regards that the context of the phenomena under study is experienced in similar ways by all actors in the same context (Orlikowski & Baroudi 2002). The majority of researchers have rejected the premises of positivism, claiming that objective and subjective realities are not mutually exclusive, no absolute source of knowledge exists, a single true reality cannot be expected, and findings cannot be true, nor is inquiry value–free. This resulted in the evolution of post-positivism. The post-positive approach presumes that reality exists but is imperfect (Guba & Lincoln 1994). **Critical** research analyzes critically the status quo to change the social conditions and existing social systems and reveal any contradictions and conflicts that may be inherited (Orlikowski & Baroudi 2002). **Interpretive** research is characterized by observations and findings which are not considered factual or objective, but it supposes that people are assumed to behave in a similar situation in similar way.
Interpretive research sees reality as socially constructed and the researcher as revealing reality (Walsham 1995a, 1995b). This work is an interpretive study.

**Case study research**

Case study methods are widely used because they may enable access to data which cannot be accessed any other way (Rowley 2002). According to Eisenhardt (1989:549), case studies are “Particularly well suited to new research areas or research areas for which existing theory seems inadequate”. Typically, multiple sources of data are used in case study research, such as documents, interviews, artifacts, and observation. The number of units is typically less than in survey research, but the depth and extent of details for each unit is great (Rowley 2002). As stated by Yin (2009), the case study approach is suitable for research questions such as Why? and How? and this method can be used for explanatory, descriptive and exploratory research (Rowley 2002, Yin 2009).

The work at hand is an interpretive case study. An interpretive approach involves understanding the phenomenon under study subjectively. The criteria of interpretive research aim to increase the understanding of the phenomenon in its cultural and contextual situation by studying it in its natural settings and from an actor’s or participant’s perspective, where the researcher does not impose an outsider’s a priori conception on the situation (Orlikowski & Baroudi 2002). The major source of data collection of an interpretive study is interviews (Walsham 1995a), and these are used in this study.

The case study method has both strengths and weaknesses which are acknowledged in the literature. The case study approach is often viewed as lacking rigor and objectivity, and this sets a specific demand for very detailed articulation and description of the research design and implementation. The case study method is seen as appropriate when a researcher tries to examine a “contemporary phenomenon in the real-life context, especially when the boundaries between the phenomenon and the context are not clearly evident” (Yin 2009:13). Yin (2009) notes a poorly defined data analysis process as one of the weaknesses. The case study method requires 20–30 individual interviews in the context of the phenomenon under study (Creswell 1994).

Cunningham (1997) proposes three different types of case study: intensive, comparative and action research. Intensive case studies do not rely on one particular type of data collection method. The intensive case study method suggests that an understanding of real settings with a range of explanations or
interpretations is developed. Evidence from different viewpoints and time perspectives needs to be addressed by the researcher. Comparative case studies stress the use of contrasting observations from different settings and emphasize the development of clear concepts. Cunningham presents three comparative analyses of cases: *case survey*, in which large amount of cases are studied and tabulated using common factors or categories; *case comparison*, in which an explanation is created for one or several cases, then replicated so as to make comparisons and provide explanations based on it; *action research* focuses on doing research through intervening and observing processes of change in order to understand and solve a problem or issue.

A case in general can be stated to be the object of a study (Runeson & Host 2009), to be analyzed using the selected method so that the unit under analysis plays a role, too. In many qualitative studies, the case method is applied in such a way that a number of cases is defined, such as particular projects, products or organizations, and both within-case and across-cases analyses are carried out. For the former, the unit of analysis may, for example, be a certain type of actor or activity.

In this research, the object of the study is stakeholder interactions in mobile software productization. This ‘case’ is analyzed based on information gathered from interviews, company documentation, project reports and participant observation: in other words, from one case company. The informants selected can be found in Appendix 1 and the company documentation in Appendix 3.

However, the analysis is done with the help of the a priori model, using the model and its elements as the units of analysis, rather than, for example, by analyzing several different productization projects separately and then cross-evaluating the results. Therefore, the study is case research in the general sense and related to acquiring the data from one case company, whereas the data is analyzed using the a priori model and no cross-case analysis is done with any data gathered from different companies.

**Action research**

Lewin (1946) introduced the term action into research, by combining the creation of a theory with a changing social system, in which the researcher is acting. Since then, action research has been characterized by many scientists. For example, Eden and Huxham (1996:75) indicate that when following it, “the research output results from an involvement with members of an organization over a matter which
is of genuine concern to them”, and (Frost 2002:25) states that it is “a process of systematic reflection, enquiry and action carried out by individuals about their own professional practice”. Therefore, the key characteristic feature of action research is that it is established on relationships between the researcher and those being researched: the objects of research become partners of the researcher during the research process, so that people are seen as "co-creating their reality through participation, experience and action” (Denzin & Lincoln 1994:206).

From a teleological perspective, action research tries to formulate situation-specific insights rather than universal laws, by producing a model or a paradigm and a series of events and activities (French & Bell 1999). Typical outcomes of action research are thereby solutions for problems and learnings from outcomes, which form contributions to science and knowledge (Coughlan & Coghlan 2002). Another way to state this would be that action research does not result in empirical implications, but that reality is incrementally constructed through the research.

It is obvious that features of action research are included in this study, through the researcher's active involvement in the case organization's operations such as gathering, analyzing, producing and making use of information that is in part applied as research data. However, the true iterative cycles of analyzing the proposed changes based on the data were not carried out by the researcher in the case organization as a part of the study. Moreover, the initial or at least the sole purpose of data gathering was not scientific research. For this reason, it is fair to state that the research stems from an action research-oriented setting, with a heavy involvement of the researcher in improving the case organization's practices, but that the reflective part of the study depicted in this study was undertaken later and did not result in yet another change in the case organization.

Despite this, it can be claimed that the data analyzed in this research was produced through time and successive changes in the case organization. Therefore it is not only a snapshot, but reflects the evolution of the empirical context of the study. In particular, the researcher was neither absent from these developments, nor just a participant observer, but was herself actively involved in the process, taking part in actions and witnessing their results. In other words, she has a knowledge of the target phenomenon under study similar to that possessed by action researchers.
Data collection

Data is collected from multiple sources in case study research. The use of multiple sources of data corroborates the same fact or phenomenon (Yin 2003). The sources are typically interviews, archived records, artifacts, observation, and participant observation. Rowley (2002) presents three principles for data collection:

1. Triangulation. All data from multiple sources are corroborated to support the same fact or finding.
2. Case study database. A database in which all data from multiple sources is collected and saved to provide transparency of the findings as well as helping to organize the data.
3. Chain of evidence. This must be maintained by a researcher, and appropriate citations to the data should be traceable.

Triangulation is seen as a major strength of case study collection and useful in minimizing the biases of the researcher and the researched (Remenyi & Williams 1996). The data analysis is based on investigating and categorizing the data (Rowley 2002). The case study protocol guides the data collection. Rowley (2002) identifies the following three steps to be followed in case study research.

1. Overview of case study project
2. Field procedures, such as use of different sources of information
3. Access to these sources.

The case study guides the interviews and formulation of questions, or the questions that the case study researcher needs to keep in mind when collecting data. It is suggested that when potential data sources are selected, the main focus should be on understanding which data sources will best help to understand the case. In a case study, a substantial amount of data is gathered informally beyond the formal data collection, not only during a certain period but in a continuous process during the research. Case study research provides a mass of accumulated information, and the most critical task is to get rid of most of the data collected and to be able to focus on the most important data (Stake 1995).
Validity and reliability

Validity and reliability are used commonly in the qualitative research paradigm (Golafshani 2003). Reliability is considered to be the “extent to which a measurement procedure yields the same answer however and whenever it is carried out; ‘validity’ is the extent to which it gives the correct answer” (Kirk & Miller 1986:19). According to Yin (1994), reliability is about demonstrating that the operations of the research (e.g. data collection procedures) can be repeated, leading to the same results. A case study is an “empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident” (Yin 1994:13). A case study is suitable for the research that finding answers to ‘how’ and ‘why’ questions about a contemporary phenomenon within its real-life context (Yin 1994). This study tries to understand interactions and how they influence the software productization phase. The informants of the case company could be interviewed again.

The trustworthiness of qualitative research is often questioned by positivists, due to the fact that validity and reliability cannot be addressed according to their own naturalistic views (Shenton 2004). Guba (1981) proposes criteria for pursuing trustworthy research. The trustworthiness of the research can be assessed through credibility, transferability, dependability, and conformability. Credibility is about assessing whether the findings represent a credible interpretation derived from original data. Credibility is one of the most important factors in establishing trustworthiness. Dependability (in preference to reliability) is addressing the quality of integrated processes of data gathering, data analysis, and theory generation. Conformability (in preference to objectivity) indicates how well the gathered data supports the research findings (Shenton 2004).

According to Yin (2003) the quality of empirical qualitative research is evaluated through four criteria: construct validity, internal validity, external validity and reliability.

Construct validity can be achieved through two measures. Firstly, triangulation should be used to view the same phenomena from different angles by using multiple sources of evidence. Secondly, the clear chain of evidence is emphasized to allow the reader to reconstruct how the research process has been carried out from initiation of research questions to the conclusions (Yin 2003). In the study at hand, the construct validity relies on using multiple sources of evidence. Interviews, project reports, participant observation, company
documentation, and the author’s personal work diaries were used as sources of data collection. The chain of evidence was achieved by using a research database and documenting the details of the interviews and the company documentation used in the study.

*Internal validity* demonstrates causal relationships between variables and results. A pattern-matching technique was carried out in this study by matching the empirical data with the pattern developed in the a priori model (Chapter 5).

*External validity* concerns the degree to which the findings and conclusions of the study can be generalized beyond the immediate case study. Case studies rely on general analytical generalization, not on statistical generalization. The researcher strives to generalize a particular set of results to a broader theory (Yin 2003). External validity could be improved by conducting this research in other companies in the same industry.

*Reliability* can be achieved by using the case study protocol and study database in the case study context, ensuring that the research procedures are documented and the study can be repeated by another investigator (Yin 2003). Reliability was aimed at by keeping a research diary and using NVivo software to analyze and classify the empirical data in this study.

**Empirical data collection**

The case company is a multinational global company operating in the telecommunications industry. The company’s products are mobile devices including software. The author was a member of the case organization acting in operational mode development tasks, and this gave me a good opportunity to achieve trust in the case company among the informants and other case organization community members.

Empirical data was collected within one case organization from different sources: interviews, company documentation, and participant observation. This aimed to increase the credibility and decrease subjective preliminary appraisals that might bias the interpretations. Company documentation and participant observation were used to support and complement the interview data. The interview data collection method was open-ended interviews, for which an interview guide was prepared. The interview guide provided a structure for the interviews and kept the focus during the interview, which was conducted in conversational mode. The interview notes and observations were written down during the interview, and the interview was transcribed as text-based notes as
soon as possible after the interview. Complementing company documentation such as current state analysis reports, lessons learned materials, memos, and project status presentations were collected in parallel with the interviews. The data collected was organized and documented in a case study database to enable other investigators to review the evidence and not to be limited by the case study report. The use of a case study database increases the reliability of the case study (Yin 2003). Yet another way to increase reliability is to maintain the chain of evidence. This means in practice that an external observer, for example the reader of the case study, can follow the derivation of any evidence ranging from the research question up to the conclusions (Yin 2003).

**Interviews**

Interviews are the most common sources of evidence in case studies, because case studies often concern human interactions and affairs (Yin 2009). Interviews can be used as tools for obtaining multiple realities of one single case by receiving different players’ interpretations in the case (Stake 1995). The interviews of case studies are often open-ended, allowing the researcher to ask for facts and opinions, leaving space to pay attention to interviewee’s body language, voice tone, wording, expressions, and other non-verbal behavior.

The interviews were carried out during spring 2011 as shown in Appendix 1, and they typically took around 1.5 hours each. An interview typically started with a short introduction given by the researcher about the purpose of the study, confidentiality, notes being made and how the results would be presented later to all informants and the management team. An interview guide, as presented in Appendix 2, was used to maintain the focus of the conversation. The researcher made notes during the interview. The questions in the interview guide were generated for the context of the interview. The atmosphere of the interview was conversational, partly because the researcher works in the same organization in operational mode development tasks, and the informants and the researcher had long been familiar to each other, and partly because of the resultant flexibility, allowing unforeseen topics to emerge during the conversation, giving valuable information for the research. A final question asked the informant if there was anything else they thought should be covered. At the end of the interview, the next steps were walked through. When all of the interviews had been conducted, all informants were sent an email to express appreciation for their cooperation and reminding them about the next steps. The interviews were conducted on the case
organization’s premises, as well as via teleconference and net meetings. Thirteen interviews were conducted face to face and seven via net meetings. A complete list of interviews, including their date and length is presented in Appendix 1.

The informants were selected from different sites of the case organization and were from Finland, the USA, Denmark and the United Kingdom. Informants were in a managerial position within the same business unit, but in different organizations within the business unit involved with software productization. The justification for selecting informants among managerial positions was to ensure that the informants had extensive work experience in the case organization and a holistic understanding of product development, and additionally they were at least partially responsible for software or hardware deliveries to product development projects. In a large global company, software developers often have limited visibility on holistic cross-functional product development activities.

The total number of informants was 20, and their average company experience was 15.7 years. The longest company experience was 22 years and the shortest was 10 years. Seven interviews out of twenty were conducted in Finnish, and thirteen in English. Notes in interviews conducted in English were written in English, and notes in Finnish interviews were written in Finnish and translated into English immediately after the interview. The translation was done as precisely as possible, taking into account the author’s then current knowledge of English. The informants were asked questions regarding organizational interfaces, cooperation, information exchange and availability, cross-functional decision-making and visibility of decisions and plans in the case organization. The questions were open by nature, allowing the informants to describe their opinions and thoughts about the question or topic freely. Interviews are characteristic of qualitative research aiming to gain the thoughts and experiences of the respondents (Cresswell 1994). The interview guide is presented in Appendix 2.

The interviews amounted to a total of 118 pages of transcribed text. The transcripts of the interviews were not written word for word, but as sentences on issues and items that the informant spoke about in the context of the interview questions. Because of the conversational nature of the interview, those sentences and comments were left out which were not in the scope of the interview guide. Handwritten notes were typed up as soon as possible. At the time of the interviews, the company was experiencing a very turbulent period. Unfortunately, news about reduction of employees, outsourcing or closing down some sites was received nearly weekly at the time. The atmosphere was not very confident due to this uncertainty and it was typical for employees to be cautious with their words.
and comments. Considering the exceptional circumstances was seen as appropriate by the author. The author wanted to ensure that the interviews were considered confidential and informants were able to express their opinions without fear of being caught later.

**Documentation**

Documentation is useful for assuring details and further providing supportive details to corroborate information from, for example, interviews (Stake 1995, Yin 2009). Yin (2009) reminds that additional documentation should be viewed critically because it does not present the absolute truth about the topic concerned, yet the researcher can easily become prone to thinking like it.

Company documentation related to programs and projects, for both the organization and the teams, was collected. The documents included internal documentation such as process descriptions and project reports. The total number of process-related documents analyzed totaled 347 pages, and the project reports 191 pages. The company documentation is listed in Appendix 3.

**Participant observation**

Participant observation is a mode of observation in which the researcher is participating in the events being studied. Being a member or a key decision-maker in an organizational setting in the case provides good conditions for participant observation. The author participated in the events under research, being involved with daily meetings, reviews and the actual work and kept a personal working diary resulting in 294 pages of notes.

According to Yin (2003), participant observation creates unusual opportunities such as being able to access events that would be otherwise inaccessible to scientific research. Another benefit is the ability to perceive reality from inside. Participant observation has been criticized as hindering production of an accurate portrayal of the phenomenon under research. Participant observation may expose or provide an opportunity to manipulate events. Whether or not manipulation occurs, the use of interviews and documentation assumes a passive researcher. The biggest problem related to participant observation is potential bias. The researcher is not able to work as an external observer, and sometimes may assume positions or advocacy roles against good scientific practice. The second concern is that a researcher may become a supporter of the organization being
studied. Finally, the role of participant observer role may stress the participation by undermining the observer role. Observational evidence is considered useful in providing an additional source of information for the study. Relevant behavior or conditions of the research focus environment can be used for observation serving data collection as one source of evidence. Typically direct observations can be made throughout meetings or during interviews (Yin 2009).

6.3 Research process and empirical data analysis

The research process and the analysis of the empirical data are described in this section.

6.3.1 Research process and empirical data analysis

The research process in this study involved interplay between data collection, literature review and the a priori model developed. The a priori model is based on literature review. The literature review was carried out in autumn 2011 and spring 2012. The literature review was started by screening and identifying articles from the research problem. The a priori model was developed for this research as shown in Figure 7, and the literature inclusion and exclusion criteria were determined by its themes. The more detailed and iterative literature search and review was performed for the selected areas based on the four themes of the a priori model: stakeholders, interactions, productization and mobile software development. Along with the literature review, the studies eligible for this research were identified and focused.

The literature search was done by using keywords and strings such as:

- Stakeholder related literature: stakeholder, stakeholder management, stakeholder involvement, stakeholder cooperation/collaboration
- Interactions related literature: interactions, communication, collaboration, cooperation, cross-functional product development, cross-functional R&D
- Productization related literature: productization, software/service/product productization, commercialization, mass customization, new product development process, innovation management, cross-functional product development/product management
Mobile software development related literature: mobile/portable device software development/engineering, mobile, mobile software, mobile software development, software product management

The sources of literature review (books and articles, proceedings of the conferences) search included for example:

- ACM Digital Library
- IEEE Electronic Library
- Science Direct
- Wiley InterScience
- Springer Link

The journals used included such as: International Journal of Project Management, Information and Software Technology, Journal of Managing Projects in Business, Journal of Product Innovation Management, Project Management Journal, and R&D Management, which were especially relevant for this research.

A typical case study research process consists of the following steps (Yin 1994):

1. Design the case study
2. Conduct the case study
3. Analyze the case study evidence
4. Develop conclusions, recommendations and implications.

The research process applied in this study is presented in Figure 8.
First, the research problem was defined (Chapter 1), and the a priori model was developed (Chapter 5) based on a literature review (Chapters 2, 3 and 4). A case organization was selected because the researcher was working there at the time. The selected case organization is an industrial company which operates in the telecommunication business, and involves both product development and software development. The case company also applies a software productization approach as part of their product development.

The research was designed as a single case study and the data collection done via interviews. Notes were written in the interviews, and rewritten in more detail after the interview. An analysis of the data was conducted based on the preliminary framework (Chapter 7), and discussed (Chapter 8), and finally a conclusion was drawn and the research questions answered (Chapter 9).

### 6.3.2 Analysis of the empirical data

The interview data was given more weight than participant observation and company documentation. The unit of case study was an organization. A coding method was used to analyze and transform the empirical data into findings. The data collected was transferred to an NVivo- database (QSR 2014) in order to manage the chain of evidence for the research findings. The a priori model
developed in Chapter 5 was used to guide the empirical data structuring. The maintenance of a formal research database provides the possibility to review the evidence and findings supporting the study report. This increases the reliability of the research (Ellram 1996). The software can be used to store source files such as textual transcripts, graphical presentations and notes, and to code and categorize the data. One of the major concerns of case study research are the validity and reliability of the results from the large amount of data collected (Eisenhardt 1989). According to Yin (2009), the use of an organized database for qualitative data increases the reliability of the research. The database enables visibility for other investigators to trace the findings leading to the conclusions. The list of main codes is shown in Appendix 4. The steps of the analysis are described next.

First level analysis. The interviews captured as transcripts, and the findings from company documentation and participant observation were transferred to the NVivo- research database. The coding of the data was started after all of the interviews had been conducted. Open coding was used first to segment the empirical data into smaller meaningful pieces. During this phase, the software was used to highlight words and sentences to identify the meaningful data. The data was first assigned to relevant themes which were given descriptive names corresponding to the structure of the interview guide.

Second level analysis. After the first level analysis, axial coding was conducted to group the open codes into the categories related to each other and to the research framework as presented in Chapter 5. The purpose of this phase was to expand and complete the coding on a more detailed level, review the data and combine the data from the first analysis into overarching concepts. The data was broken into more manageable parts while maintaining the links and traceability between the parts and the original entity. The research database software allows one to highlight and assign those findings which are assumed to be important for the research questions. In particular, the Boolean contextual search enabled by NVivo was useful in order to find certain words in the documents saved in the research database. Each node was given a name that described its content, for example certain quotations were included in the code “Project start and initiation”. Defining the nodes was dynamic; the codes were reviewed and refined as new meaningful segments were found in the empirical data. Around five rounds of coding were done. New codes emerged as the researcher found interesting connections.

Third level analysis. The purpose of this phase was to organize the data in order to address passages to the research questions. Open coding was used for
data that did not fit any existing category (node) so as to handle data on a more detailed level.

*Fourth level analysis.* The results of the analysis were compared with the preliminary framework, and the implications of the results were discussed.

### 6.4 Limitations of the research

The study at hand represents a qualitative study based on interviews, company documentation, and participant observation. Software productization was studied in one case organization. As the case organization is large and global, the results might have to be verified in a smaller or national company. The fact that one case organization was analyzed does not demonstrate that the results can be in generalized in all companies developing software products.

Aspects of action research were included, but no improvements were introduced to the case organization due to the fact that the case organization does not exist at the moment in the form that it took at the time that the empirical data was gathered.

Few proposals to be considered in software productization were generated by the research process. Interactions were broken down into three elements: whether they existed in the case company, were to be developed or had to be improved. The elements identified in the research framework were identified and relevant in the case organization. Interactions were focused on software productization perspectives, not on other stakeholders’ perspectives.

### 6.5 Summary

In this chapter, the research design, research process and analysis of the empirical data have been introduced.

This study is a qualitative interpretive case study. Epistemologically, this study is post-positivistic and ontologically transactional and subjectivist. The research was conducted as single case study, and the typical case study research process applied. Triangulation was applied, as the data collection was done from multiple sources such as qualitative interviews, company documentation, and participant observation. The main focus of the data collection was in interviews. The interviews were guided by an interview guide. Validity and reliability were considered throughout the research process. The data was analyzed by using a coding method, and the results are introduced next.
7 Analysis of stakeholder interactions in mobile software productization

This chapter analyzes the empirical data collected in this study. Chapter 6 described the research methodology and design used to analyze the data. This analysis is based on the framework presented in earlier chapters. The analysis of the data is based on the a priori model presented in Chapter 5. The empirical data is presented in this chapter.

The empirical data of this study consists of qualitative interviews, company documentation, and participant observation. The sources of empirical data are cited as codes as described in Table 8.

<table>
<thead>
<tr>
<th>Codes (examples)</th>
<th>Type of source</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1, I8</td>
<td>Informant</td>
</tr>
<tr>
<td>N2, N6</td>
<td>Company documentation</td>
</tr>
<tr>
<td>P1, P9</td>
<td>Project report</td>
</tr>
<tr>
<td>D1, D2</td>
<td>Author’s personal work diary</td>
</tr>
</tbody>
</table>

The case organization and background of the mobile software development are introduced in Section 7.1. Cross-functional software productization related empirical data is analyzed in Section 7.2. Stakeholders of cross-functional software productization related data are described in Section 7.3. Stakeholder interactions in cross-functional software productization are analyzed in Section 7.4. The interactions can be broken down and categorized as communicative, collaborative and integrative interactions. Software productization is recognized as a phase during product development in which an extension consisting of differentiated software features and functionalities is added on top of the core mobile product to create a final touch to a product to be brought to selected target market or customer segment.

7.1 The case organization

Nokia is referred as the case organization, which was part of Nokia Corporation at the time. It operates in the telecommunications industry. Nokia is engaged in producing mobile devices and telecommunications services and equipment. One of its main strategies has been in introducing new disruptive technologies and
solutions for consumers and markets. It produces complex high-end mobile
products to the market, but during the past decade the importance of inexpensive
mobile devices for emerging markets (for example India, China and Africa) has
become increasingly high.

Nokia is one of the leading mobile manufacturers in the world, and its market
share of the mobile handheld market was 23.8% in 2011 (Gartner 2012), as shown
in Table 9. The average personnel count in Nokia in 2011 was about 51,404
employees (Nokia homepage 2011).

Table 9. Market shares of mobile device vendors 2011 (based on Gartner 2012).

<table>
<thead>
<tr>
<th>Company</th>
<th>2011 Units</th>
<th>2011 Market Share (%)</th>
<th>2010 Units</th>
<th>2010 Market Share (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia</td>
<td>422,478.3</td>
<td>23.8</td>
<td>461,318.2</td>
<td>28.9</td>
</tr>
<tr>
<td>Samsung</td>
<td>313,904.2</td>
<td>17.7</td>
<td>281,065.8</td>
<td>17.6</td>
</tr>
<tr>
<td>Apple</td>
<td>89,263.2</td>
<td>5.0</td>
<td>46,598.3</td>
<td>2.9</td>
</tr>
<tr>
<td>LG Electronics</td>
<td>86,370.9</td>
<td>4.9</td>
<td>114,154.6</td>
<td>7.1</td>
</tr>
<tr>
<td>ZTE</td>
<td>56,881.8</td>
<td>3.2</td>
<td>29,686.0</td>
<td>1.9</td>
</tr>
<tr>
<td>Research in Motion</td>
<td>51,541.9</td>
<td>2.9</td>
<td>49,651.6</td>
<td>3.1</td>
</tr>
<tr>
<td>HTC</td>
<td>43,266.9</td>
<td>2.4</td>
<td>24,688.4</td>
<td>1.5</td>
</tr>
<tr>
<td>Huawei</td>
<td>40,663.4</td>
<td>2.3</td>
<td>23,814.7</td>
<td>1.5</td>
</tr>
<tr>
<td>Motorola</td>
<td>40,269.0</td>
<td>2.3</td>
<td>38,553.7</td>
<td>2.4</td>
</tr>
<tr>
<td>Sony Ericsson</td>
<td>32,597.5</td>
<td>1.8</td>
<td>41,819.2</td>
<td>2.6</td>
</tr>
<tr>
<td>Others</td>
<td>597,326.9</td>
<td>33.7</td>
<td>485,452.0</td>
<td>30.4</td>
</tr>
<tr>
<td>Total</td>
<td>1,774,564.1</td>
<td>100.0</td>
<td>1,596,802.4</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Nokia was divided at that time into four business units: 1) Mobile Phones, 2)
Smart Devices, 3) Location & Commerce, and 4) Markets. Nokia’s structure is
presented in Figure 9.
Mobile Phones’ mission was to produce “modern and affordable mobile experiences to people around the world. In particular, the team leverages its innovation and strength in growth markets to bring people affordable access to the internet and applications and – in many cases – provide them with their first ever internet experience” (Nokia homepage 2011).

Smart Devices focused on smartphones. The business unit delivered Symbian with new models and software updates and aimed to strengthen the product portfolio with the Windows Phone platform (Nokia homepage 2011).

Location & Commerce developed “integrated social location products and services to customers, as well as platform services and local commerce services for device manufacturers, application developers, internet service providers, merchants, and advertisers”. This business unit was also a “leading provider of comprehensive digital map information and related location-based content and services for mobile navigation devices, automotive navigation systems, internet-based mapping applications, and government and business solutions” (Nokia homepage 2011).

The Markets business unit focused on “selling Mobile Phones products, executing winning marketing and communications, creating a competitive local ecosystem, sourcing, customer care, manufacturing, IT and logistics across all Nokia products” (Nokia homepage 2011).

Nokia Siemens Networks was jointly owned by Nokia and Siemens. According to the Nokia homepage (2011) it was described as follows: “It is one of the leading providers of telecommunications infrastructure hardware, software and professional services globally”.

Fig. 9. Nokia’s organizational structure (based on Nokia homepage 2011).
Nokia is referred to as the case organization in this study due to simplicity.

7.2 Cross-functional mobile software productization

Mobile product development projects take the overall responsibility for delivering the productized product to fit the predicted market window for maximum profit. Each mobile product development project is responsible for the product from the development until it is ready to be manufactured and launched to the market. In software productization, the mobile software product is integrated from core software components provided by the software development platform and new product specific software components implemented within software productization (N10, N12).

Software productization is initiated when the first product release is received from software platform development. The software project manager takes responsibility for leading software product development with software productization (N10). During this phase, a number of new product-specific requirements and needs are received from stakeholders, which are entered into the software requirement management system and prioritized (D2). Fundamental software changes at this stage, for example affecting architectural designs, are not desirable: they should be included in the Product Release created by software platform development. A Product Release is an integrated product specific software package consisting of reusable core software components enabling new key features and functionalities. Software productization is about implementing the final product-specific software features on top of Product Release for the defined market segment. The software manager is responsible for the software productization phase, ensuring that the software product meets product-specific targets. The software productization phase also includes the validation of the software product to meet market/customer requirements, company targets and interoperability with agreed standards (N10).

Numerous new needs and requirements are addressed by multiple stakeholders to software in productization. The software project manager is the hub of interactions, balancing all of the software needs and requirements coming from different stakeholders and being responsible for the mobile software product (D2, N12). The conceptual model and the modularity of software productization are presented next.
Conceptual model of software productization

The conceptual model of software productization can be considered as a planning phase in which the concept of the mobile software product is defined at the beginning of the software productization phase (N10). The software product concept is a detailed description of the Product Release derived from the product abstract defined in a product roadmap (N12). At this stage, the necessary new core software components are identified and defined, i.e. the gap between the Product Release and market segment or customer-specific features. The target of software productization is to ensure high-quality product data and on-time product delivery. The purpose of software productization is to configure and integrate the actual mobile software product from core software product consisting of core software components delivered by software platform development. Software productization is responsible for developing the product-specific software components and adding them on top of the core mobile software product. It is the software manager’s responsibility to investigate and define the gap between the core mobile software product and new product-specific requirements received from different stakeholders, turning them into new requirements to be implemented in the final product (N8, N10).

A mobile software product consists of core software components including the basic software technology and features. Mobile product-specific software components are developed during software productization within a mobile product development project, and these new software components are added on top of the core product. The gap between the core mobile software product and the necessary new software components for the specific market must be identified at the beginning of software productization. The software manager in a software productization within a mobile product development project is responsible for analyzing the gap and creating a plan for the new software components that are needed. The role of product portfolio planning and management activities is essential for successful software productization because it has ownership of the product (N10). The success of productization is measured through:

1. Process efficiency (milestone slippages, planning accuracy and on-time-delivery, and project costs)
2. People measures such as project team climate
3. Process output metrics such as failure rates
4. Productization metrics such as error and trends, open item counts and trends (N10).

Software productization also includes the validation of the mobile software product to conform to defined user needs or requirements, and agreed interoperability with standards, services and other products. The success of the software productization phase is critical in order to launch high-quality product to the target market in time and to improve the business performance of the company (N10).

**Software productization through modularity**

Software productization through modularity can be considered as an execution phase in which the implementation of new product-specific software components is done. The software product data and configuration rules are also established. Software productization is part of mobile product development, and it takes place with considerable amount of freedom and independence. All mobile products within the same product family share the same software technology, common software features and properties which can be seen as a core part of the mobile software product. In software productization, a set of differentiated product specific features and functionalities are developed and added on top of the core software product by a single product development project (N10).

Software productization starts when a first product-specific software release is delivered to software productization within a product development project. The delivered software release consists of core software components. Software productization is responsible for developing the product-specific software components which are added on top of the software release delivered by software platform development. The main goal of software productization is to integrate and stabilize the product software (N10).

Software release planning and requirement management is an essential part of software productization in order to deliver the mobile software product in time to the market (N3). This has come under continuous improvement activities during the last two decades in the case organization, and during recent years, lean and agile methodologies have been applied in software release planning and requirement management as part of software product management (D1). The software development community has adopted improved practices well, but there are still some issues with non-software related organizations. There are mutual
dependencies between mobile software and hardware, and when hardware changes are made, the consequences often cumulate in software development, causing unexpected software development work. Informants told that the visibility of hardware platform development plans was not adequate, and hardware changes often came as a surprise to software productization (I2, I10), as Informant 2 said:

“Sometimes new requirements mean changes to hardware and original design, and sometimes hardware changes are done and nobody tells the software teams about the changes, the modified hardware does not work without changes in SW. They just assume that hardware change does not have anything to do with software, hardly so.” (I2)

The product concepts are the basis for initiating software product development. Product and portfolio management is responsible for defining and communicating the product concepts. The concepts are on a high level, indicating the mission and the market segment of the product. Product concepts are broken down into more detailed software requirements to be developed by software platform development. Software platform development provides the enabling software technology and major functionalities to software productization, which will take the responsibility for finalizing the product in order to meet the product concept (N3). The product concepts seemed to be received too late in software platform development to ensure sufficiently mature and timely software delivery to software productization (I12, I13).

7.3 Stakeholders of software productization

Mobile product development is a cross-functional activity in the case company consisting of a complex network of stakeholders and their functions. Software productization is the final phase in mobile software product development, and it is loaded with stakeholder expectations, needs and requirements, transmitted through numerous interactions. Considering stakeholder expectations is a crucial element for strategy implementation (N12). The most important stakeholders of mobile software product development are marketing, corporate management, product and portfolio management, platform development and product development (I16, I17, N10) as shown in Figure 10.
Software productization can be considered as a project within the mobile product development project. A software project manager is responsible for the software productization, and plays a key role in interacting with multiple stakeholders and addressing their expectations and needs (N10). The stakeholders, their key functions and interactions with mobile software productization are introduced in the following sections.

**Marketing**

The products are meant to meet market needs and attract new customers. Marketing is at the front line to explore and identify, understand and capture market needs, future trends and opportunities. It captures and processes market information, establishes market segments, and feeds to the corporate management in order for them to make strategic choices and decisions about investments in future technologies and products (I10, I14, N6, N16). The role of marketing is to create needs for innovations and new technologies in the market. It transforms the identified market opportunities into product requirements (N16). Marketing information is integrated in corporate management, and product and portfolio management decision-making so that they can make strategic decisions about the company vision and long-term product strategy. The outcomes of the integration are product roadmaps and the master backlog (N6, N16).
Marketing is visible in software productization mainly through transmitted marketing presentations, user experience targets, market- and product-specific forecasts, customer feedback, and new needs and requirements of the market and customers (N6, N16). As marketing information is integrated in product and portfolio management, the market needs are embedded in the roadmap and master backlog. Marketing allocates a product manager to each product development project. They are the representative of marketing and have continuous discussions with the most important current or assumable the most important future customer of the mobile product under development (N16). A product manager is part of the product development steering group (N11). In that sense, marketing is integrated with product development and the collaboration becomes true in daily or weekly mobile product development meetings. A product manager is responsible for creating a more detailed product concept for the mobile product and they continuously analyze the market and the status of the product under development. The product manager ensures that the customer’s voice is heard in a specific product development project and they play an important role in discussions related to planning product functionality and features, keeping in mind the strategic goal of the product. The product manager is active with major customers, such as mega-operators, during the product development project and may forward some requirements to software productization directly from customers.

“Priorities may come from individual top managers, sometimes from customers through the product manager. Mega-operators’ deal closing may depend on some feature and that is impacting on our priorities, if there are deals to make, this impacts on priority.” (I1)

As stated by Informant 1, this may lead to confusion in the prioritization of requirements.

**Corporate management**

The most important task of corporate management is to create a long-term game plan for shareholder value creation. It is the highest authority in defining the business that the company is in and the choices of priorities for investments and trade-offs. Corporate management is responsible for developing, translating, and implementing the strategy. Strategy implementation consists of communication, management, and follow-up. The formal communication of strategy, vision and mission is achieved through internal company information-sharing sessions,
company internal digital marketing material, and information-sharing guides for middle managers, encouraging them to leverage strategy discussions at the team level (N4). The strategy is based on the company’s own vision and market information that has been gathered. The capability to integrate marketing information in corporate management decision-making is crucial. The company strategy and visions are made visible via product portfolio management producing business unit level roadmaps consisting of future projects, technologies, and services (N16).

Corporate management has the authority to decide what kind of products would be developed (I1), as stated by Informant 1:

“The business unit leadership team decides what programs are started, and they are defined in portfolio presentation and it is updated monthly.” (I1)

Corporate management also has the authority to approve and reject product development project milestones (N12). The strategy is translated into products by product and portfolio management and it is communicated to all other organizations through roadmaps and the company-level backlog; the so-called master backlog. The roadmap consists of a long-term plan for products and their visionary schedules. The master backlog includes all products, future technologies, opportunities, innovations, and major functionalities or features of products in prioritized order. The roadmaps and master backlog are an integrated joint effort and an outcome of the collaboration of corporate management, marketing and portfolio management (I1, I10, I15, I16). Corporate management is entitled to present new requirements directly to on-going product development project and software productization. New requirements and needs are sometimes received via emails and phone calls and the formal processes are bypassed (I2, I6, I10). The priorities of this kind of needs are also sometimes stated via email.

“... Prioritization is done based on product information coming via emails, not based on use of backlogs.” (I10)

This was considered a problem due to the need for reprioritization of the software requirements in the master and team level backlogs (I2, I11, I20).

Product and portfolio management

Products to be developed are defined by product and portfolio management (N12). The case company applies product family model based product and portfolio
planning and management. Innovations and new technologies are typically implemented and introduced to the market in a lead product. Copy (variant) products within the same product family are built on top of the lead product technology, but are usually designs of their own made from cheaper materials and with reduced functionality. Utilizing the matured previous assets for copy products ensures that existing assets are reused as such, and no further development or verification is needed. Reusing assets leads to faster product development, improved quality and reduced risks (N12).

The product and portfolio management process (PFMP) is one of the key processes in Nokia (N12). Product and portfolio management is the informed decision-making process matching the needs and capabilities to maximize value, balance risks, and align strategy. The outcome of product and portfolio management is a strategic roadmap in which the market slots, platform products, and systems are combined. The decision-making of product and portfolio management is based on market pull, technology push, and the strategy and constraints of the company (N11). Another important output is the master backlog, which is the prioritized list of products, key features and technologies (N18).

Vast amounts of market data about trends and market needs are processed in marketing and fed and integrated to product and portfolio management. Product and portfolio management combines both the market needs and opportunities and the future technological choices on a time line indicating the long-term strategic plan of the company (N11). Product and portfolio management is responsible for creating and maintaining the company product portfolio roadmaps and prioritized master backlog, as well as for evaluating products and projects and prioritizing them (N16). A roadmap consists of product families, including technology and business opportunities, market slots, platforms, key features, systems and mobile products, and takes the form of a strategic scheduled map (N5). A product roadmap is a presentation consisting of product families, including lead and copy products on a timeline, as shown in Figure 11.
A master backlog is a prioritized Excel list of all items mentioned in the roadmap; single technologies, key features, and functionalities are also included. The roadmap and master backlog serve as vehicles for communicating the long-term visionary plans across the company. The roadmap and master backlog are updated monthly in order to reflect the corporate’s fast response to changes in the market, operational environment, and new available technologies (N14). The Product Portfolio Management Process (PFMP) is a long-term planning and decision-making process to maximize business value, balance risks, and align the strategy. In a PFMP, the products and projects are evaluated, selected and prioritized. The PFMP is also responsible for transforming market and user needs into technical descriptions. The role of product and portfolio management is to accelerate, slow down or terminate product development projects, and to balance resources accordingly. Product and portfolio management looks proactively at disruptions, technology and the market in order to react to changes but bears in mind the corporate strategic goal. PFMP is a decision-making process which creates the strategic roadmap and guides the execution of Nokia product creation (N11).
Product and portfolio management communicates the roadmap and master backlog to the relevant organization managers, who communicate the information further in their own organizations. This was considered as one-way communication by development organizations such as software platform development and software productization, and the need for a more collaborative way of working was raised by informants. According to informants, it was not clear how products, technologies, and major features are selected into the roadmap and master backlog, what the justifications for the items and their priorities are, and who makes or participates in the decision-making process regarding new items (I10, I15, I16):

"Prioritization is not clear or used as a method, this is not known of prioritizations .... There are conflicting priorities, different stakeholders want everything, nobody is giving up. The highest manager gets what he/she wants ... no prioritization rules exist." (I10)

Informants felt that they did not have any chance or even channel to give proper feedback or comment on the content of the roadmap and master backlog (I2, I3). They felt it to be unfair that their commitment was required by corporate management and product and portfolio management as soon as the new versions of roadmap and master backlog were published, no matter how conflicting the priorities or how loaded their resources (I2, I10, I20).

"Capacity planning scenarios and request for input to those comes typically by first, by email. They want our commitment even if we do not know what has been requested from us: it is difficult. We are asked how much effort we are going to put into the technical feedback (in studies and feedback for other functions) and implementation and planning." (I2)

The roadmap of product and portfolio management is an input for the Product Creation Process (PCP) which plans, develops, and maintains products and related process capabilities ordered by the PFMP. Product and portfolio management work is undertaken together with corporate management and marketing, and cooperation is integrated in their processes (N16). The need to fill product and service portfolio gaps with new needs and opportunities increases the pressure to find solutions for the efficient management of the product portfolio. PFMP is an incentive to start a new product development project and software productization. PFMP reflects the key stakeholder expectations. Each product
included in the roadmap is supposed to have an initial product concept and business target i.e. the key theme of the product and target market (N11).

Mobile products are planned within device families which consist of one lead product and several copy products or variants. Products within the device family are derived from the same common product software platform. All products within the same device family share the same platform and technology basis, meaning that there is a key technology or key selling feature for each device family (N8). A new technology is built in R&D in platform development and is delivered first for the lead product, in which form it is introduced to the market. The copy products (variants) using the same software platform are usually targeted to be cheaper and with less functionality (N2, N8). Product and portfolio management makes decisions reflecting the strategy of the company and changes in the operational environment. Decisions about major features per product and selected technologies to be applied are also made during this phase (I17, N17). New technologies and innovations are mapped for the product families, and the key selling points are mapped for each lead and copy product. The product roadmap provides visibility of the product implementation plan up to 18 months ahead, although the accuracy is more blurred the further ahead products are mapped. The product roadmap is reviewed once a month by the product and portfolio management team, and updated accordingly. The product portfolio is based on the company’s business opportunities, and market and customer needs. New opportunities may also emerge unexpectedly, and the resulting portfolio product gaps must be filled quickly (N10, N11).

In general, it is accepted that in the telecommunications industry, the pace of change is fast and frequent and the company must react quickly. Agile and lean methods applied in product development embrace the flexibility and capability to adopt quick changes. Based on the data, the visibility of decision-making process regarding new items added to the roadmap and master backlog was considered deficient (I8, I11, I15).

**Software platform development**

Finding a more efficient and disciplined way of developing mobile software has been a focus area for years in Nokia. Software platform development has been established in order to respond quickly to changing market and customer needs with more efficient and predictable software production targeting to timely software deliveries (I3, I9, N7, N8). The purpose of software platform
development is to deliver reusable and configurable software technology assets to ensure competitive advantage by creating solutions which are easily integrated and maintained (N8). The latest methods, such as agile and lean methods, have improved prioritization and planning practices in software development communities in Nokia, in which backlogs have been used in order to help project managers and developers to focus on the relevant areas (N9, P1). Individual products share common software technology within one product family. Improved cost effectiveness and higher productivity is gained through software platform development (N8). Additionally, the software development risks are seen to be minor, due to learning and diminishing the rework. The software platform way of working enables more effective mass customization of products (N12). Software platform development is responsible for delivering modular and configurable core software components in Product Release. It is developed by software platform development software teams (N7, N13). Software platform teams’ development work is initiated by the product and portfolio roadmap in which the mobile product concept, i.e. key theme or product abstract, is defined. The master backlog provides master guidelines for the priorities of the products, features and technologies to be implemented in software platform development (I2, I3, I8).

Software platform development provides technological solutions and basic software capabilities and functionalities to product development project. Software platform development is initiated based on the roadmap and master backlog created by product and portfolio management (N9). Software platform development is responsible for transforming software-related requirements in a software technology or software-based system that enables the products to meet customer needs. It creates reusable and configurable core software components which are integrated into software productization. Additionally, software platform development studies new technologies and opportunities and develops and creates pilots of innovations (N8).

Agile and lean values and principles are applied in software platform development, but there are still multiple software development processes in use: sequential, incremental and agile ways of working. In a sequential development process, the software development is done sequentially, aiming at one software release per corresponding mobile development project. Incremental software development aims to make cyclic software releases to produce software functionalities in increments. Agile methods stress incremental and iterative software development aiming at short cyclical software iterations (N8, N9).
There were issues in software platform development with the prioritization and synchronization of multiple level backlogs (I1, I8).

“Our #2 item on our backlog may be #45 in the master backlog or vice versa ... and implementing teams have their own backlogs, they have many products to serve, and products are competing, priority is not clear. On a high level it has not been solved how to manage that in different functions because they serve multiple products at the same time. It depends on the program manager how strongly he/she drives the implementation work.” (I1)

The master backlog is the highest level backlog, stating company-level priorities reflecting the corporate management targets (I8, I16, I20, N18). In addition to multiple product releases for on-going product development projects, the software development platform is responsible for studying and piloting new technologies and opportunities and introducing them as commercial solutions for products in the product roadmap. These studies are carried out as individual projects but are not often shown in master backlog; instead they compete for the same resources allocated to projects listed in the master backlog (I15, N19). The effort required for new technology studies is significant: the lead time of new software technology from the study phase to commercial implementation may take up to 4–5 years (I1).

**Product development**

All mobile products are developed in product development projects. Product development is guided by the product creation process (I1, I10, N12). PCP is a cross-functional effort, i.e. a collection of competences to achieve the targets of the company. The output of PCP is the productized product delivery consisting of the product data/recipe, offering capability and delivery capability. PCP defines the process of mobile software development as part of mobile product creation. A product development project consists of cross-functional teams establishing a collection of different necessary competencies. The milestone model shown in Figure 12 is the main tool for synchronizing and aligning the product development phases of cross-functional development (D1, N12).
Fig. 12. Milestone model applied in product development (based on N12).

The product development project is responsible for creating the single product targeting to the agreed schedule, planning and preparing the demand and supply network capacity for the manufacturing, and ensuring that the product is ready to be delivered to customers. A mobile product development project consists of several technology area sub-projects which synchronize their plans according to the product’s project milestone plan. According to PCP, product competitiveness should be built in the planning phase, and changes made during the development phase will delay the deliveries (N12).

Mobile software product development is guided at a high level by the product and portfolio management roadmap which is the key document stating the product development schedule and key selling theme(s) or feature(s) for each mobile product, for both lead and copy products within the same product family. The priority is also stated in the master backlog, but in addition to products, that also includes prioritized technologies and single features. In practice, mobile software development is done by dedicated software teams in software platform development and in mobile product development projects. They are also responsible for the software release and requirement management. The software
platform development team is responsible for delivering the mobile product-specific product software, Product Release, to a product development project for the product family (N13). Product Release consists of hardware-integrated and tested core software components forming the basic software product release for the targeted mobile product. After Product Release delivery, the software manager for the product development project will take responsibility for the further development of the Product Release as a market segment-specific mobile software product. Mobile software development during the product development project is focused on adding product-specific software components on top of the core software release, i.e. Product Release for certain market segment. This phase is referred to as software productization (N10, N11).

Software productization is responsible for creating the final software product, targeting to agreed schedule, planning and preparing the demand and supply network capacity for the manufacturing, and ensuring that the software product is ready to be delivered to the market (N10, N11). The case company applies agile and lean principles at all levels of the organization. Although the milestone model is not considered a central element for measuring the readiness of the software product, it is still the major instrument for measuring the progress of the product development (I3, I5, I18).

The development of mobile software is undertaken according to the software engineering process. This is among the key processes, and its purpose is to transform a set of software-related requirements into a software product or software-based system to meet customers’ needs (N7, N9). The core software capability is developed in software platform development. The priority of the work to be done in software platform development is stated in the product roadmap and prioritized master backlog (N11).

Summary

Software productization is a phase in which the product-specific software layer is added on top of the core mobile software product. The outcome of software productization is a finalized software product delivered to the market. Software productization is influenced by many stakeholders, and as such it is a cross-functional effort. A software project manager is responsible for interacting with stakeholders and considering and balancing their needs and requirements. The identification of the most important stakeholders with the majority of the influential power and interest is crucial in order to satisfy their needs and create
successful software products. Based on the data, marketing, corporate management and product and portfolio management are the most influential stakeholders influencing software productization because of their managerial position, authority and responsibility for strategic planning. An overview of the stakeholders of software productization and their key functions is presented in Table 10.

Table 10. Summary of stakeholders of software productization and their key functions.

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Stakeholder key functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marketing</td>
<td>Creating and communicating company strategy and vision</td>
</tr>
<tr>
<td></td>
<td>Decision-making about starting or canceling mobile product development</td>
</tr>
<tr>
<td>Corporate management</td>
<td>Creating and communicating company strategy and vision</td>
</tr>
<tr>
<td></td>
<td>Decision-making about company resource allocation</td>
</tr>
<tr>
<td></td>
<td>Decision-making about starting or canceling mobile product development</td>
</tr>
<tr>
<td>Product and portfolio management</td>
<td>Reflecting corporate management strategy and visions through roadmap and master backlog</td>
</tr>
<tr>
<td></td>
<td>Accelerating, slowing down or terminating product development projects</td>
</tr>
<tr>
<td></td>
<td>Balancing resources</td>
</tr>
<tr>
<td></td>
<td>Defining and updating monthly the product portfolio roadmap and company level master backlog</td>
</tr>
<tr>
<td></td>
<td>Defining product abstracts and key features of the products in the product portfolio</td>
</tr>
<tr>
<td></td>
<td>Maintaining long-term plan for product families and individual products (lead and copy products) and technologies applied</td>
</tr>
<tr>
<td>Software platform development</td>
<td>Creating software technology enablers for product development project (software productization)</td>
</tr>
<tr>
<td></td>
<td>Investigating and studying new technologies</td>
</tr>
<tr>
<td></td>
<td>Delivering core software components to product development project (software productization)</td>
</tr>
<tr>
<td></td>
<td>Software error correction in collaboration with product development project (software productization)</td>
</tr>
<tr>
<td></td>
<td>Delivering tested and integrated Product Release to product development project (software productization)</td>
</tr>
<tr>
<td></td>
<td>Software release and requirement management</td>
</tr>
<tr>
<td>Product development project</td>
<td>Creating the mobile product offering capability to the market</td>
</tr>
<tr>
<td></td>
<td>Software product release and requirement management</td>
</tr>
</tbody>
</table>

Examining the informants’ responses, the term stakeholder was familiar, but there was no description or evaluation of any systematic stakeholder mapping used to
identify the stakeholders or their importance, influence and priority. Some informants were uncertain when pointing out their stakeholders.

Marketing plays an important role in software productization. A product manager is allocated to the mobile product development project, and is responsible for creating a detailed product concept and for observing the target market and passing changes and new customer needs from the single product point of view for the software project manager. Market information is fed to corporate management who has the overall responsibility for product strategy. Corporate management, in cooperation with marketing and product and portfolio management, create a product roadmap indicating the target schedule of each mobile product, and a master backlog to guide the software implementation work in software platforms and software productization on a more detailed level by showing the priorities of products, technologies and key functionalities/features to be implemented. The software platform creates the core mobile software technology for the product development project. Software productization is a project within a product development project which is responsible for finalizing the product-specific mobile software product for the target market. The main tool for synchronizing product development and organizing cross-functional work is the milestone framework.

7.4 Interactions in cross-functional software productization

Mobile software product development is a cross-functional effort consisting of the competences needed to achieve the target. Software productization is at the focal point of balancing the priorities and needs of stakeholders and co-teams (I1, I19, N12). The elements of stakeholder interactions in cross-functional software productization are introduced next.

7.4.1 Communication

Communication across the company is considered as a major element of good collaboration and a crucial factor for creating relationships, which in turn are fundamental elements for creating successful products (I9, N14, P6). Visibility is an important part of communication (I3, I4, I8, P6).

“There is good stakeholder communication between functional areas, special expert teams and hardware and software platforms, meaning that people are
working together, though we still miss some cross-functional visibility on all relevant parties’ plans and statuses. We have wikis, though.” (I3)

Communication is conducted through information exchange between different stakeholders providing mutual visibility via documentation and common tools (D2, I5, I7, N14). The major channels in providing visibility are documentation, different types of meetings, information meetings, and planning and review meetings, and providing access to software tools or systems in which the needed information can be accessed (D2, N1, N12). Communication through information systems or documentation is not enough, because there is a risk that the documentation could be misunderstood, especially if the documentation is inadequate, out of date or missing (D1, I17). Although visibility is provided, it may be targeted to the wrong audience. Regular and continuous cross-functional discussions among stakeholders support the common view gained through information systems and documentation (I11).

Marketing plays a key role in gathering and understanding market and customer needs. It follows market trends, creates market- and product-specific forecasts and identifies new business opportunities. The gathered information is translated into more clearly defined market/customer and product requirements. Product market forecasts, business opportunities, and requirements are combined as key drivers for product development (N8). The needs of marketing, corporate management and portfolio and product management are stated in the product roadmap and master backlog (N11). The processed market information is integrated into corporate management decision-making. The outcome of marketing and corporate strategy is combined in the company product roadmap and master backlog. The high-level product abstracts are seen in the roadmap. Marketing takes an initiative role in the software productization planning phase because mobile product software products are established to fulfill the product abstract. The communication is conducted mainly through documentation, chiefly the product roadmap and master backlog (N6, N11).

Marketing is present in software productization through the product manager role which is allocated to the mobile product development project. The product manager is responsible for creating a more detailed product concept derived from the product abstract, and continuously follows the product-specific market forecast and situation, ensuring that customers’ needs are fulfilled. The product manager is a member of the product development steering group and actively participates in product-related decision-making. The product manager is
responsible for detailed definition of product content and communicating market- or customer-specific needs and requirements to software productization (N6). A product manager represents marketing as a stakeholder, but new needs or major requirements often bypass the product manager, and new requirements are received directly via emails or calls from corporate level marketing. Market needs and requirements are seen to be communicated in one direction only. According to informants, the software productization team has really no or very limited means to influence new requirements received this way. The lack of mutual communication with marketing was seen to increase the risk for missing market opportunities, because marketing is a stakeholder whose approval is needed for major product features or software improvements (I9, I10, I11).

Corporate management represents the highest decision-making authority in the company. It is responsible for the communication of strategy, mission and vision across the company. Market information is integrated into corporate management decision-making. The communication is conducted through information-sharing sessions and presentations, and the company’s internal web pages are used in sharing information, usually in the form of videos and documents (N1). The corporate management strategy for the software productization planning phase is visible through the product roadmap and master backlog (N6, N11).

Corporate management may strictly control activities during the software productization execution phase. The turbulent industry in which the case company operates may require quick changes such as canceling or establishing new product development projects, decisions about resourcing and passing new major requirements to mobile software product development. The visibility of decision-making and justification of business decisions was seen as inadequate:

"Corporate management decides the programs, I do not have visibility on the decision process ... we get it only as information - like for-your-information."  
(I15)

Often sudden requirements do not follow the formal decision-making process and the fact that new sudden requirements are not stated in formal documents such as the product roadmap or master backlog increases uncertainty for the software productization project and may decrease staff commitment and motivation (I1, I5, I15). Informants felt that they were not considered or involved enough in decision-making regarding the mobile software product under development. Communication with corporate management was not considered as mutual, since
there was no formal channel to give feedback on the product roadmap and master backlog, and informants felt that a listening attitude is missing (D1, D2, I2, I20, P7).

*Product and portfolio management* is responsible for combining the corporate strategy and marketing views as a product roadmap. It also creates the high-level product abstracts, i.e. defines the key feature/functionality for each product (N8). The mobile product abstracts on a high level are stated in the roadmap created jointly by marketing, corporate management and product and portfolio management. The product abstract is the starting point for software productization planning. The prioritized master backlog is the second major document, including more specific corporate management targets and consisting of mobile products, technologies and major features or functionalities in prioritized order. Both the roadmap and the master backlog state the company strategy and visions of corporate management (N6, N11).

Product and portfolio management’s influence is seen in software productization through monthly product roadmap and master backlog updates. Monthly changes cause prioritization issues and may cause redundant work and uncertainty whether software teams are working with the appropriate features and priorities. The visibility of the product roadmap and master backlog was seen inadequate because they are delivered to only some managers, and information about changes do not reach all relevant persons or teams; new needs may therefore come as a surprise (D1, I11):

“Changes sometimes surprise me, we have not been communicated about some new technologies or features we are requested to develop.” (I11)

Sometimes new projects are secret and new items are not shown in the roadmap or master backlog intentionally. This caused concern and a feeling that software implementation teams’ expertise is not being used in product portfolio planning. However, software development teams’ commitment was required, even when the targets or requirements were not communicated. (I1, I2, I16). The needs received through the product roadmap and master backlog were clearly recognized to reflect corporate management’s objectives (I1, I13). The communication of the updated product roadmap and master backlog is conducted mainly through monthly emails to the target group (N19). Because the case organization consisted of staff of many nationalities, sometimes proper communication was weakened by the language barrier.
Communication was considered to be rarely mutual but rather one-way only, and informants felt that they were not heard in the product and portfolio decision-making process (I2, I20, N15). This was especially seen as a problem when plans were too optimistic or unrealistic regarding the number of requirements, resourcing and implementation schedule (I2, I4, I11, N15, P8), or the expectations of corporate management were considered unrealistic (I20). Informants expressed that new needs and requirements are just brought out through documents which had already been approved at corporate level. Sometimes the key theme or product abstract was not available or communicated since the products were on the roadmap or master backlog. According to the informants, some software development work was sometimes started based on assumptions when information about new requirements or functionalities had not been received or was not available early enough. This could cause redundant software development work (I7, I12, I16):

“Nowadays products come too late, the technology development is started based on guesses and the customer studies, operator opportunities, and own technology roadmaps which are not based on our top management plans and marketing requirements or our business intelligent studies.” (I16)

According to informants, more visibility of the decision-making process was desired. It was considered important to know who made the decisions and the background justifications for decisions. The lack of visibility of decisions made regarding the product roadmap and master backlog was particularly stressed (I5, I15, N15). The monthly updates of the product roadmap and master backlog were seen to be the major cause of conflicts in software product development (I10, I12). Alignment and synchronization with the master backlog and reprioritization of software team-specific backlogs was seen as a very difficult task (I2, I8, I10). New items on the product roadmap and master backlog caused uncertainty and concerns across software development organizations including software productization, since multiple software productization projects were waiting for core software deliveries (I2, I6, I8, I19).

“I have visibility on the business unit backlog but I don’t know how the prioritization is done, based on what criteria, it would be nice to know at least. Processes and tools should be clear, now there are many different... language barriers in communication when English is not the mother tongue.” (I20)
processes, and many milestone models which are still applied. How do we synch all technology areas planning and deliveries without milestones?” (I2)

The prioritization of products and features has been an issue in the case organization during the last decade and much effort has been allocated to searching for better ways to manage product development capability and resources (I10, N15, P11). Agile and lean methods have been introduced as key solutions to solve the prioritization problem, and these methods have been adopted widely at different organizational levels of the company, although the transition is still on-going (N9, N15, N17, P1). The master backlog adopted at company level is one visible outcome of the agile methods to clarify and to communicate common company priorities across organizations. The downsides of the roadmap and master backlog are that their visibility is restricted. The information cannot be delivered and communicated openly to all employees across the company, but only to a certain level of managers, because of the strategic nature of the information, which is highly confidential and considered as secret (I1, I8, I16, I20).

“They also make nowadays BU master plans (BU epic backlog) which is delivered to us monthly via email at least to managers, that is secret I think, not shared with all members in our organization.” (I1)

Information about the roadmaps and master backlog is then filtered and communicated further to the rest of the company members. The roadmaps and master backlog are updated monthly, often with very significant and quick changes. The master backlog in particular impacts highly on software platform development and product development, including software productization, by causing changes in their team-level priorities and as such was seen as a major source of conflict. In addition, information about new products in the roadmap was seen to come at very short notice, as Informant 2 said:

“New programs start out-of the blue, no warning, and it is not known by who and where it has been decided. There is visibility of the BU level backlog, the so-called master plan or backlog, but there are so many different level items on the same list, like epics, experiences, features, technologies ... The problem is that there are different priorities on different level backlogs, and teams do not know what to do, and what is the priority, they deliver to several product projects, what is the priority. I would like to say that there is no clear decision-making path for establishing a new program or project.” (I2)
According to informants, the problems regarding the multiple levels of priorities were caused by conflicts in aligning, prioritizing and synchronizing different level backlogs (I1, I11, I16, I19, P5, N15).

“I think they want to help the prioritization of the work, but it is not working too well. All the work is meant to be prioritized through this backlog but there are conflicts, for example, with our backlogs.” (I1)

The visibility of product roadmaps is about two years ahead, which was not seen adequate (I7, I10). However, creating longer-term roadmaps is challenging due to the fast evolution of technology and new innovations. The product roadmap is a highly confidential strategic plan, which makes it difficult to share the information with all members of the company. This was seen as a retarding factor for faster software development.

“Sometimes the roadmaps are even secret in some technology areas, and not visible even for those who should see them.” (I20)

The restricted visibility of the product roadmap and master backlog is justifiable, but it may also leave out some relevant stakeholders. Cross-functional visibility is necessary in order to synchronize planning and implementation activities and to provide an early view on new decisions, new features and functionalities, plans, and changes made by different stakeholders. Ensuring adequate visibility is not a simple task in a large organization. In a fast-changing environment where daily changes occur, early, up-to-date and online visibility prevents redundant work and waste of effort in a development project (I16). Early visibility of decisions leaves no room for speculation and development can be started earlier as decisions are visible (I15).

Platform development. The role of software platform development in successful software productization is crucial. A mobile software product has many dependencies on multiple platform development areas such as hardware, ready-made chipsets, suppliers and specialized software technology areas such as audio, user interface, display and video technologies (I14, I16). Software platform development is at the front line in implementing the software technology enablers for software productization, and as soon as the mobile product development project is established, communication is vivid and daily until the Product Release is delivered to the mobile product development project. The development of core software technology may take software platform development up to four years, so
their visibility on product roadmaps as early as possible is essential in order to start the development.

“Visibility should be better to other functions’ plans and especially roadmaps, because they have lots of dependencies with each other. We could react in time, if we saw the roadmaps.” (I8)

During software productization, the formal use of planning tools enables cross-functional visibility at least between mobile software development and software platform development (I3, N1, P5). Technology areas were still seen as working in silos; informants felt that the usage of different tools is not the root cause, but it was rather the case that technology areas were not willing to share their information and plans.

“There is no visibility to other functions planning and no synchronization is done together, all the time surprises happen and we try to use tools to provide visibility, but they are not used.” (I16)

This was considered a severe problem because it made synchronization of development tasks difficult and slowed down software productization (I10, I11). However, communication between software productization and software platform development was perceived as efficient, generally adequate and intensive (I13). The communication is carried out by providing cross-functional visibility of plans and in common planning and review meetings and, as such, two-way communication. Common meetings allowed participants to share thoughts and jointly solve and discuss issues (I5, I11, I13).

Product development is the context in which the software productization takes place. The communication between a product development project and software productization is seamless, as software productization can be considered as a project within a mobile product development project.

“There is good stakeholder communication between functional areas, special expert teams and hardware and software platform, meaning that people are working together, though we still miss some cross-functional visibility of all relevant parties’ plans and statuses.” (I13)

All mobile product development team members use the same tools and processes, and the communication is conducted through daily and weekly meetings and discussions (I11, N10, N12).
The need for wider cross-functional visibility to plans and progress outside product development was stressed (I11, I16). Changes happen during mobile software product development up until the very last phase. Last-minute changes which bypass formal requirement management processes were considered particularly difficult (I2, I4, I5, I20). Late phase new requirements shorten the software testing time and expose the software product to new software errors (I11, I20). Some software changes may impact on the software architecture and may require massive software development work and sometimes changes in hardware, too (I13).

“Changes are coming from various routes weekly – no consistency.” (I2)

“Some functions give their plans in a late phase, when some very principal decisions have been done already, and they should have been involved with the initial planning.” (I11)

Communication with software platform development was perceived as good, and based on cross-functional frequent meeting practices, common planning sessions, and tools that may provide visibility for plans and progress throughout the company (I4, I5, I8, I20). Unclearly defined roles and processes hinder communication and greatly affect the speed of software productization (P1, P3, P10). Because of parallel processes, it is difficult to find reliable information in software tools because they are configured to support one way of working only (N15).

Communication relies greatly on personal networks and relationships, but this was not considered sufficient. It is important that communication is regular and that visibility provided through software systems is trustworthy and up to date. A lack of a formal communication process and supportive tools were seen as issues of poor communication (I11, I16, I17). Software development activities sometimes are started based on emails, even though the visibility of the decision-making process is missing (I6, I10, I16). Formal software requirement management processes and tools are available, but not all counterpart organizations and stakeholders use the tools or the formal requirement management process, forcing the product development project manager to expend effort in managing needs outside the agreed procedure (I2, I16, I9).
7.4.2 Collaboration

Collaboration issues have been a topic for several years (P1, P4). The defined company-level product creation process emphasizes collaboration across stakeholders (N11). Information sharing increases the understanding of product development dependencies (I2). Based on the data, the involvement of all relevant stakeholders in target setting was seen as important for promoting collaboration (I4). Collaboration in general between stakeholders was perceived to be in place on a personal level but not coordinated through processes or practices (I6).

Teams were seen to work in silos due to lack of cross-functional visibility and common meetings, but the milestone model was considered a major collective tool to ensure collaboration across stakeholders to some extent (I1). Collaboration was weakened by poor or inadequate documentation and irregular communication (I8, I11). Overall the need for group work such as common planning workshops and reviews was seen a way to improve collaboration and also a means to improve general work satisfaction and pleasant working conditions (I1). The need for the early and intensive involvement of stakeholders at the beginning of software productization was emphasized (I2, I13).

Collaboration is supported by common tools and regular formal meeting practices such as common planning and review meetings (I2, I5, I15). The case company was in the middle of a transition to a new way of working. This had been the case for years, and as a result there are now functions applying different methods and processes which make it even more difficult to understand each other’s plans. Major obstacles for successful collaboration were perceived as overlapping, multiple and unclear processes, roles and responsibilities (I1, I11, I16, N15).

“We have multiple processes and methods still in use and this makes it difficult.” (I1)

“We are applying Scrum, but programs are applying milestone models. We have many processes in use at the moment, and the backlog comes from agile and lean methods.” (I16)

The definition and communication of responsibilities of new roles was seen as insufficient (I4, I13, N15).

The synchronization of the implementation work and plans was considered extremely challenging (I11, N5). The lack of visibility and communication of the justifications for decision-making in corporate management and product and
portfolio management caused major concerns in software platform development. Decision-making is often understood as major stakeholders passing over the formal agreed processes to present claims which cannot be ignored.

“Sometimes managers walk over what is suggested by the software designer, although he/she has the knowledge and knows the consequences and dependencies, or a late requirement is accepted to the backlog even when the teams and software manager say it is not possible, due to the risk of having lots of errors which cannot be corrected in the set timeframe. We should stick to major requirements agreed in the beginning and not do this and that whatever.” (I17)

When these claims come without the formal decision-making process, effort is wasted in wondering where the claim came from and whether it is valid or not. Work may be started based on rumors rather than justified and communicated facts. (I12, P9). Common face to face cross-functional planning and review meetings and workshops were seen as the best way to prevent misunderstandings and improve collaboration between stakeholders towards common goals (I8). Collaboration was weakened by restricted visibility of cross-functional plans and targets. There may be some secret projects, plans and goals which are revealed to a very small group of people. Although the projects or plans are secret, the functionality is provided by the same software development teams. This causes challenges in prioritization of resources and tasks and makes it difficult to gather a holistic picture of the company strategy. Unclear goals prevent collaboration and commitment, and this should be considered in regard to secret project goals, too. (I12, I20).

Marketing collaborates intensively with corporate management and product and portfolio management, and the most important visible outcome of their collaboration is the product roadmap and master backlog. Collaboration between marketing and software productization is achieved through a product manager who is allocated by marketing: the product manager is an active member of product development project and thereby also involved with software productization. (N6, N12). The collaboration between marketing and software productization was, however, perceived as minor (I6, I17). The product manager is responsible for preparing the detailed product concept within a product development project and focuses on monitoring the product-specific market and customer needs and changes (N11). The interaction between marketing and software productization was perceived as being based on one-way communication.
**Corporate management** leads the strategic planning of the company. The informants did not find collaborative practices with corporate management, and expressed the desire and need for collaboration. Interaction was considered to be one-way communication, and there were no agreed practices to provide feedback or comment on the product roadmap. Informants felt that corporate management was far away from the daily work, and there were no means to influence decisions made by them (I8, I13). According to informants, collaboration consisted of common meetings, mutual discussions and the ability to influence each other’s decision-making and plans, and all of that was missing (N15, N17). Sometimes new requirements were received from corporate management at a very late phase of software productization, and informants commented on the lack of transparency on decisions that had been made, and explained that if there is no visibility of the decision-making process, the collaboration is undermined (I12, I9, N15).

**Platform development.** The collaboration between software platform development and product development including software productization was considered adequate and good (I6). The interaction between software platform development and software productization is very intensive because it delivers core software releases to software productization (N8). Collaboration ideally consists of cross-functional common meetings such as cross-functional reviews and planning meetings, the usage of common tools, willingness and openness to cooperate, and good relationships between teams and individuals (I7). Communication and cross-functional visibility across plans and progress were seen as prerequisites for collaboration (I5, I11).

“Cooperation is a must, we have common release planning, and of course our milestones are synchronized with all others’ milestones. We plan together, but there is no visibility until it’s too late usually. Then there are all business unit managers involved and crash actions and weekly or daily follow-ups are started. We try to consider all players but we do not see their plans.” (I1)

According to informants, software productization plans are synchronized with software platform development plans. This was seen as an enabler for good collaboration. (I1, I3, N14).

Regular cross-functional meetings such as synchronization meetings and common monthly planning meetings were considered the most informative and beneficial. Good collaboration between software platform development and software productization was conceived as contributing to success (I4, I10, I13).
Unified working methods and processes were seen to enabling and facilitating good collaboration (I2). Parallel and multiple concurrent processes make collaboration difficult or even prevent it. The involvement of all relevant stakeholders in the product roadmap creation phases are seen as an opportunity for better products and faster software productization. (I4).

Product development. As software productization is a project within product development, the collaboration is carried out through daily meetings and informal discussions. The product development project has the overall responsibility to create the mobile product, including the software product, so software productization is in the interest of product development, in order to ensure that the software product will be finalized (N10, N12). The software platform is in an essential role to successful software productization. The collaboration between the platform development and the product development project starts as soon as the product development project is established, and will continue intensive until the product development project will be terminated and the mobile product will be in the maintenance mode (N8, N9). Along with agile and lean methods, new collaborative working methods, for example Scrum and common planning meetings, have been taken into use and these have changed the cross-functional development towards a more dynamic and proactive mode. These methods emphasize and corroborate frequent reviews in which stakeholders’ plans are aligned and discussed. (I17, N14).

Integration

Integration is enabled by communication and collaboration and in practice through the same tools and procedures, such as providing visibility of plans through documentation, using the same software tools and common planning and review meetings (I16, I19). According to the findings, the role of integration as a discrete element of interaction did not emerge as assumed and described in the a priori model. As it can also be considered as a synonym of collaboration, it is included as one form of collaboration in this study.

The integration of stakeholders in the case organization is guided by processes and procedures, for example, by the product creation process (N12) and the milestone model (N1). Each stakeholder has its own milestone definitions and criteria, and these milestones are aligned with the product development milestone model representing the highest product development steps. In practice, the integration is achieved through checklists which define the maturity and format of
each stakeholder’s deliverables (N12). The milestone framework defines all of the product creation steps, and all technology areas plans are synchronized towards the same target (I2, N2, N14, P1). The milestone model is based on stages in which relevant stakeholders provide their contribution to product development. The disadvantage of the milestone model is that different stakeholders see their contribution as linear, leading to long feedback loops between stakeholders. Secondly, the milestone model may prevent some other stakeholders from proceeding, because the spaces between milestones are defined (N2, N12). However, milestone reviews were seen as an important integrative practice improving communication across stakeholders (I10). The case organization’s internal integration happens through cross-functional team orientation linking organizational teams and functions horizontally (N2, N6, N14). Well-adopted integration across stakeholders results in good and open collaboration and communication, improves the capability to solve problems faster and enables faster feedback loops between stakeholders (I16, N14).

“Cooperation happens too in common problem solving workshops, when the right people are in the same place, problems are solved fast.” (I16)

Marketing is integrated with corporate management and product and portfolio management through formal processes to ensure that market information; trends, needs and consumer feedback are considered and included in corporate strategy and product strategy decision-making (N6, N16). A product manager allocated by marketing is one formal and visible act of integration in product development project and software productization. The product manager ensures that the product-specific market and customer needs are considered in the finalized version of the mobile software product and observes the product-specific market forecasts. That information is fed back to corporate management, product and portfolio management and the product development project at hand. The product manager’s role in software product content definition is crucial (N8, N12).

Corporate management is integrated with marketing and product and portfolio management through key processes (N2, N4, N6, N11). The integration with platform development and product development is carried out through the product strategy reflected in the product roadmap and master backlog. Corporate management authority is visible also when it makes decisions regarding starting, canceling or reprioritizing products and development projects. (I2, I10, N12).

Product and portfolio management integration with corporate management and marketing is ensured through key processes, for example product creation
process. The integration with product development through is visible in documented processes, but in practice, the activities of product and portfolio management are seen through product roadmap and master backlog. According to product creation process, product and portfolio management representative is a chairman of the steering group of the product development project. The steering group ensures and supervises firstly, that the product-specific business targets are met, secondly, that customer needs are fulfilled, thirdly, that the product at hand fits the company product portfolio, and fourthly, that there is sufficient capability and capacity to invest in the development. (N12).

Platform development integration with corporate management and product and portfolio management seemed to be very thin (I2, N15, N17, N20). Platform development collaborates with the product development and software productization within as soon the product development project has been established. The finalization, planning, validation and error correction of the Product Release is done together in close cooperation and the support from platform development continues until the product development project has been terminated and is in maintenance mode (N8, N12).

Product development. The product development project has the overall responsibility for the mobile product delivery. Software productization is a project within a product development project, and as such is fully integrated with it (N12). In practice, the product development project configures and integrates the mobile product from product components of either software or hardware. It is the product development project’s responsibility that the product meets the set needs and targets. The collaboration (and integration) between software productization and the product development project is seamless, as they share exactly the common goals. (N10).

7.5 Summary

This chapter has focused on empirical data and findings. The case organization was introduced in Section 7.1. Cross-functional software productization was discussed in Section 7.2. Stakeholders and their main functions in software productization in the case organization were described in Section 7.3. In Section 7.4 stakeholder interactions in the case organization were analyzed.
8 Discussion

This chapter summarizes the findings of this research from theoretical, empirical and managerial perspectives. First, the revised model of stakeholder interactions in cross-functional software productization is presented. The model is followed by theoretical and managerial implications. Finally, speculations for future research are given in the summary of this chapter.

8.1 Stakeholder interactions in cross-functional software productization

Analysis of the case data shows that stakeholder interactions in cross-functional software productization are essential, as teams or groups in organizations do not work in isolation (Bourne & Walker 2005), but are inter-related in trying to work towards common goals (Hogg et al. 2012) by exchanging information and generating knowledge (Berthon & John 2006). The a priori model identified stakeholders involved with software productization and the elements of interactions. The revised model combines the results of theoretical and empirical analysis. The a priori model was used to establish a model to investigate stakeholder interactions in software productization. The completed a posteriori model is presented in Figure 13.
Software productization is a critical element and predecessor for successful software product entry by developing the product in order to maximize the customer benefits. During software productization, an extended software product is developed and added on top of the core software product. The core software product is created and delivered by software platform development. The core software product includes the basic software technology and major features and functionalities common to all mobile software products within a product family.

Software productization can be divided into two phases, the conceptual model of software productization and software productization through modularity. The conceptual model of software productization is the phase that focuses on planning the mobile software product. Mobile software product planning is initiated by a product abstract provided by product and portfolio management. The planning phase of software productization aims to identify the gap between the core mobile software and product-specific software by gathering and defining new needs and
requirements to be included in the extended software product. No such new software requirements impacting on core software and causing reverse software engineering should be accepted at this point, only those software requirements which can be developed as separate software components to be configured and integrated with the core software during software productization. However, this basic rule is neglected sometimes when major software requirements are received from influential stakeholders. Software productization through modularity is considered as an execution phase in which the software product is implemented. During the execution phase of software productization, the software development should be focused on the implementation and integration of those software features and functionalities that were defined in the planning phase.

**Challenges of software productization**

Software development processes and methods have been the focus of operational development in the case organization for years. The fact that software development is applying agreed processes in software engineering, and that the requirements are managed systematically in software systems and many metrics can be produced to ease the follow-up, has led to an even more disciplined way of working and made the work in software productization visible and traceable. Conflicts and defects arise when formal processes are overridden by stakeholders. Processes are useful, but they are worthless if they are not applied by all stakeholders.

New major needs and requirements emerge during both the planning and execution phases of software productization. They are received through the formal software requirement management process but also directly from stakeholders via emails and phone calls. When formal processes and decision-making are overtaken, this causes confusion in software productization. The stakeholder who originates the requirement is clearly identified as empowered by managerial authority and is entitled to present new requirements. Firstly, the confusion and uncertainty arises in software productization because there is no evidence, visibility or traceability in the decision-making process, justifications and formal approval of the requirement. Secondly, prioritizing the software product-related requirements will become a problem. How should the reprioritization be done, is there any software requirement which should or can be left out, and who has the ultimate responsibility for approving the reprioritized list with new items on it?
Evidence has shown that new major software requirements are sometimes received at a very late phase of the execution phase in software productization. It is generally accepted that new software requirements may emerge at a late phase during the software product development, but the requirements should not be such that they cause a need for reverse engineering in software platform development or changes in software architecture. Some requirements are received late because the product concept has been delayed. When this happens, software development is sometimes started based on assumptions, because the biggest concern of software productization is that the product does not miss the market window. It is common for software development to be considered a bottleneck for mobile product development and a major cause of product delays, and software developers within software productization want to do their best to prevent that perception. New major requirements received outside the formal processes and at a late phase lead to the need for rework in software productization, sometimes changes in software architecture and reverse software engineering in software platform development. Delays with one software product also have impacts and increase the risk of delays to the next planned software products.

New needs and requirements are not always intentional, but may be generated by careless hardware changes. According to the data, the dependencies between hardware and software are not considered early enough, or are sometimes forgotten. For example, a change in the display size of a mobile product may require many software changes. As this leads to unexpected extra work for software development, additional effort and resources are also needed for testing, validation and integration. It seems to be typical that the overall software product development time is not extended, but the length of testing, validation and integration is shortened in order to launch the software product in the target market window.

Software productization is a critical phase for getting the mobile product to the market, and there is high pressure to shorten the software productization time. As the data shows, software development, release planning and requirement management processes are applied in a disciplined way in software productization. It would be important to trace and analyze the root causes of delays in software productization. The visibility of the impacts of major new and late phase software requirements, hardware changes and overridden processes in software productization would be revealing. The impacts are understood more clearly when they are turned into currency and working units, which are usually very
easy to comprehend, and induce the significance for the business for all stakeholders.

8.1.1 Interactions

One of the largest challenges in software productization is balancing all of the needs coming from different stakeholders through various routes. Interacting with multiple stakeholders and trying to consider their needs is not straightforward. The most important stakeholders of software productization are marketing, corporate management, product and portfolio management, platform development, and product development. Software productization is influenced by multiple stakeholders via interactions which are carried out through communication, collaboration and integration. An overview of stakeholder interactions and their intensity is illustrated in Figure 14. Intensity is demonstrated by the weight of the arrow. The heavier the arrow, the more intense the interaction. The heaviest arrow demonstrates interaction which consists of communication, collaboration and integration. Medium arrows consists of two elements of interaction, for example between marketing and product development. The lightest arrow indicates a weak intensity of interaction. Weak intensity means that only one element of interaction is identified between stakeholders. The direction of the arrows indicate whether the interactions are mutual or one-way only.
The stakeholder interactions illustrated in Figure 14 are introduced in more detail next.

**Communication, collaboration and integration**

The major element of communication is established through visibility across plans, progress and status, either through documentation or software tools or both. The importance of personal networks and relationships is seen as one of the key elements of communication and collaboration. The usage of common software systems provides cross-functional visibility and support communication. However, many informants did not consider the information in the systems to be reliable, due to their incoherent use by functions. The trustworthiness of the information provided creates trust between parties and diminishes the need for continuous discussions. In the case organization, communication was perceived to be generally good in terms of good atmosphere, openness and willingness to
communicate. The major challenge regarding communication was considered to be the lack of access to and visibility of cross-functional plans and progress enabling good communication and adequate collaboration. Providing visibility does not guarantee good communication, but only establishes one-way communication, leaving the receiver without a means to react and give feedback efficiently. The lack of visibility was explained by informants in that functions and organizations still tended to work in silos and use different software tools, blocking cross-functional visibility. In some cases, the silo working came up relation to some stakeholders not wanting to share information or restricting visibility to a few persons. Documentation and the disciplined usage of information systems supported communication but did not detract from the need for face-to-face common meetings.

Regular cross-functional reviews and planning meetings were conceived to be the best tools for progressing good collaboration and enhancing a good and open working atmosphere. Insufficient roles and unclear responsibilities cause impediments for collaboration, cause concern and increase uncertainty. Lack of transparency of decision-making processes and accountability were considered deficient in the case organization. However, the working environment encouraged and empowered teams and individuals to collaborate, even though in many cases this relied on personal activity and networking.

According to the findings, integration is seen as a synonym of collaboration and associated with collaboration. Integration is seen to corroborate collaboration and is ensured mainly through processes and the milestone model applied in mobile product development by all parties involved in product development. Milestone model product development consists of multiple lower-level organization-specific milestone models. In practice the product development milestone model synchronized all product development-related functions.

Stakeholder interactions

The most noticeable observation from the data is that software product development-related interactions are divided into two stakeholder groups. The first stakeholder group consists of marketing, corporate management and product and portfolio management. This stakeholder group can be nominated as the management group. All elements of interaction are used: communication, collaboration and integration. Integration is mainly carried out ensured via processes. The most visible outcomes, indicating intensive interactions, are the
monthly updated product roadmaps and master backlog in which market
information, corporate strategy and product planning information is combined.

The second stakeholder group consists of software platform development and
software productization, and can be called the software development group.
Accordingly the interaction among these stakeholders is intensive, consisting of
all elements of interaction. As soon as a product development is established and
all relevant roles are occupied in it, intensive collaboration is established.
Intensive collaboration is enabled by two-way communication, with mutual
visibility provided through software systems and internal company intranet pages
or Wikis. Collaboration between software platform development and software
productization was perceived to be sufficient, although the lack of cross-
functional visibility of some technology areas (e.g. hardware development) was
stressed by some informants. It may be that hardware and software development
are so different by nature that a natural joint area is not easy to find until the
software is integrated with the hardware in the product development project.
Collaboration in the software development group becomes real through regular
common meetings which can be status checks, reviews of plans and progress,
release and requirements reviews and planning and jointly held problem-solving
workshops. Integration is in place, as the output of software platform
development is the starting point for software productization within a product
development project and Product Release is finalized in close cooperation as soon
as the product development project has been established. The collaboration
continues until the product development project is terminated.

The interactions between these two stakeholder groups are mostly
communication based, and the informants perceived this as mostly one-way
communication. The role of corporate management is seen as critical for
successful new product development (Cooper & Kleinschmidt 1995), to ensure
the environment, resources and capabilities and to avoid a dysfunctional state
(Souder 1981). Corporate management is responsible for acting as sponsors of
projects and for maintaining the commitment of product development
communities. It can be questioned whether this is possible despite the lack of two-
way communication, collaboration and integration, and if the interaction was
more intensive, would that progress more efficient product development? Both
software platform development and software productization are in receiving mode.
The communication is conducted in the form of top-down information-sharing.
The monthly updated product roadmap and master backlog are received via email.
Although software platform development plays a crucial role in developing the
mobile software product by creating the basic software technology enabling the mobile software product development, it appeared that chances to influence the product roadmap were considered very limited or even missing. For example, the software platform has a crucial role in software product development by implementing the core software technology on which the mobile software product is built. The more mature the quality and functionality of Product Release, the shorter the software productization phase and time to market.

Challenges of interaction

Informants clearly expressed frustration at the lack of formal processes and channels to give their comments or feedback to the management group. They felt that their opinions were not asked at any point and they could not express their views on the content of the product roadmap and master backlog, even in cases where plans or schedules were considered unrealistic or too aggressive. The product roadmap and master backlog as such were seen as useful and informative, guiding all organizations towards the common goal. New high-priority products in the roadmap and technologies and major functionalities in the master backlog cause confusion and increase uncertainty. Software platform development develops multiple core software components for multiple product development projects simultaneously. The reprioritization of products or addition of new ones into the roadmap may risk the maturity of committed product releases delivered by software platform development and accordingly lead to increased software development effort being required in software productization. As collaboration is enabled by mutual communication, based on the findings this is very weak or does not exist. Integration is documented and defined in process descriptions, but no guidance is provided as to how to implement them in practice.

There was a lack of two-way communicative, collaborative and integrative activities between marketing, corporate management, product and portfolio management, and software productization. There were no defined processes or practices for giving feedback to marketing, corporate management and product and portfolio management in order to contribute, discuss and influence product abstracts, needs, prioritization, and requirements. The case organization has been undergoing a major operational mode transition for years. The organization is global and complex, and the transition happens at different time in different organizations. During the transition phase, organizations use partly new and partly old processes. Roles and responsibilities may be unclear within the
organization, but this also creates a major obstacle for counterpart organizations. This makes communication and collaboration even more difficult.

It was seen that the product roadmap and its schedule, and the master backlog with its prioritization information, were based on one-sided view, and software development expertise was not considered in decision-making. Particularly in the case where the development resources were fully occupied, negotiations and collaboration were desired for reprioritizing the software development team’s backlog. Informants felt that when new requirements came in, nothing was left out, and the reprioritizing of features/requirements was not conducted together with the higher authorities. Sometimes marketing, corporate management and product and portfolio management directly contacted software productization to express new needs and requirements, bypassing the formal processes. Not using the formal practices caused confusion and uncertainty in understanding the definite mandatory set of software requirements to be implemented, and which software requirements were optional or could be left out in order to launch the software product.

8.1.2 Stakeholders of software productization

Stakeholders are considered as groups which are crucial for the success of the company (Freeman 2002) and can be divided into external and internal stakeholders (Deetz 1995). Previous studies have shown that internal stakeholders influence software productization in many ways. Software productization interacts with multiple stakeholders and receives, captures, considers, and balances their needs and requirements. To fulfill and understand their expectations and the weight of attention that they should be paid, it is crucial to identify the most influential stakeholders. The identification of stakeholders in the case organization was fairly fuzzy, even though the term stakeholder was familiar to informants, and corporate management and product and portfolio management were acknowledged as the most powerful internal stakeholders. Some projects and teams underwent stakeholder identification, but it was not done systematically, and the influence and power of stakeholders was not evaluated in the case organization. Because of the turbulent environment in software productization, the identification and classification of stakeholders is essential, to know in what respect the demands and claims should be treated. It is not possible or even necessary to satisfy all stakeholders equally (Marcoux 2003, Phillips 2004). Weerd et al.’s (2006a) software product management framework presented
in Section 2.4.3 was used in the a priori model of this study. According to Weerd et al. (2006a), software product management consists of portfolio management, product roadmapping, requirement management, and release planning functions. Functions can be considered as stakeholders in the model. The findings show that product roadmapping is carried out in product and portfolio management. Requirement management and release planning are carried out by two stakeholders, platform development and product development. The revised software product management framework is presented in Figure 15.

Fig. 15. Revised software product management framework.

According to the data, the stakeholders involved with software product management are divided into two groups consisting of the same stakeholder groups as in presented in Figure 14, namely the management group and software development group. The arrow between the two boxes demonstrates how the software development group seems to perceive that software product development-related information comes as given without channels to negotiate or to be heard by the management group. The stakeholders and their involvement and functions regarding software product management are introduced next.
Marketing is an important source of information in which market needs and requirements are gathered for corporate management decision-making (Heurise et al. 2012). It explores market and customer opportunities and needs and new trends, and makes market and product or technology forecasts. Market needs and opportunities establish a basis and justifications for future product and technology evaluation (Jantunen & Smolander 2006). The marketing view is present in the product development project through the allocation of a product manager. The product manager ensures that the product meets the market segment needs, and in some cases customer-specific requirements. The product manager is focused on product launch related activities. In the preliminary model, marketing expressed its needs in a product roadmap and master backlog. However, it turned out that marketing passes needs and software requirements via a formal requirement management process or directly via email or phone to software productization.

Corporate management is responsible for creating the company strategy, mission and vision and enabling resources as mentioned by Weerd et al. (2006a). It relies on market information in its decision-making. Its product development-related objectives and goals are expressed in the product roadmap and master backlog, which are created in cooperation with marketing and product and portfolio management. Corporate management represents the highest authority of decision-making and is entitled to reprioritize, establish or cancel product development projects (Leenders & Wierenga 2002).

Product and portfolio management is jointly responsible with marketing and corporate management for orchestrating the product roadmap in which the marketing information and company strategy are combined. It evaluates products and technologies and prioritizes them. The roadmap consists of product families with a lead and one or more copy product. Typically, a product roadmap covers two or three years ahead. Product and portfolio management is also responsible for defining the product abstracts for each product in a roadmap. The product abstract is a product-specific key feature, functionality or technology. Product and portfolio management also creates a master backlog consisting of products, key features or functionalities and technologies in prioritized order. Both the product roadmap and the master backlog are updated monthly. In the preliminary model, software product management consisted of portfolio management, product roadmapping, requirement management and release planning. In the case organization, product and portfolio management is responsible for the roadmapping function by creating a product roadmap. Requirement management
and release planning is conducted in platform development and software productization. 

*Platform development* creates the enabling software technology for software productization. The development is implemented in technology teams and the work is started based on the information in the product roadmap and master backlog. The focus of platform development is to provide a core software release, which is further developed in software productization as a finalized software product for the market. Two major responsibilities of platform development are requirement management and release planning.

*Product development* has overall responsibility for creating a finalized mobile product for the market. Software productization is embedded in the product development project. According to the findings, software release planning and requirement management are done both in platform management and in software productization within a product development project. Platform development is responsible for the core software product development, including software requirement management and software release planning, until the Product Release is delivered to the product development project. Software productization within a product development project takes responsibility for the development and management of the extension part of the software product.

### 8.2 Theoretical contributions

The main purpose of this research was to study stakeholder interactions in cross-functional software productization in the context of mobile software development. A literature review was conducted on mobile software development, productization and interactions to build a theoretical model presenting the main theories and connections in the study. This model was used to guide the empirical data collection and analysis in order to answer the research questions. The findings have theoretical implications which are discussed next.

#### Software productization-related findings

Productization is a predecessor for product entry into the market (Flamholz 1995, Simula *et al.* 2008). A product can be divided into core product and an extending part. During productization, an extended layer is added on top of the core product to respond to customer needs (Simula *et al.* 2008). Productization can be achieved through modularity, in which a product is created from separate components by
mixing and combining them in different ways in order to create multiple products (Hänninen et al. 2012). Based on the data, software productization can be divided into two phases: the planning phase and the execution phase. The planning phase corresponds to the conceptual model of software productization and the execution is carried out through modularity, which consists of an integrated mixture of core and extended software components. Planning includes the software product concept which is derived from the product abstract in the product roadmap. The product concept consists of the key feature or key functionality of the product and information about the target market or market segment.

During software productization, the gap between the Product Release and the software product-specific needs and requirements is identified. Successful software productization is achieved through release planning and requirement management processes. No new major software requirements should be accepted during software productization. Major software requirements are considered to be requirements which have an impact on software architecture and cause reverse engineering and reworking in software platform development or major unexpected workload in software development teams in software productization. Product-related needs and software requirements are generated by stakeholders, mostly by marketing, corporate management and product and portfolio management. It should be noted that software requirements are not always generated intentionally, but can be caused by hardware changes, due to their mutual dependency with software. These dependencies should be considered early enough or in sufficient detail in hardware development.

The release planning and requirement management process is a central factor for successful software productization. Informants advised that new software needs and requirements are received via formal processed but also via emails and phone calls from stakeholders. Stakeholders represent managerial authority and power, and their claims cannot be ignored. However, the problem is that visibility, traceability, and justification of decision-making is missing, and the software project manager must balance between formally approved and these non-formally-approved software requirements. Due to a lack of formal decision-making, the pile of software requirements increases but no previous requirements can be left out. New software major requirements, hardware changes, and overridden processes increase the risk of delayed product launch.
Stakeholder-related findings

This study showed that stakeholders play a key role in software productization. Previous research acknowledges the importance of stakeholders’ influence on company strategy (Ylä-ranta 2006) and organizational decision-making and implementation (Donaldson & Preston 1995), performance and success (Clarkson 1991, D’Aveni 1995, Hillman & Keim 2001, Joyner & Raiborn 2005). It is important to identify the relevant and most important stakeholders, considered to be those who can affect or be affected by the achievement of the organization’s objectives (Freeman 1984).

The term stakeholder was unclear for some of the informants even though they were very experienced and had a long working history in the company. Stakeholder mapping was undertaken in some product and software development projects, but this practice was not systematically used in the case organization. It is not enough just to identify the stakeholders. It is also important to identify real names as representatives within a stakeholder group in order to involve the appropriate people. As stakeholders are crucial for the success and performance of organization, it is equally important to understand their power and influence on the organization or team in order to satisfy their needs and expectations. All stakeholders should not be treated equally and they have multiple roles and interests. It should be noted that stakeholders’ interests may vary along the time and phases of software productization. During the interviews, the informants mentioned that stakeholders such as marketing, corporate management and product and portfolio management use their managerial authority, and their claims and their requirements were not questioned even though in some cases the needs conflicted or competed, or the formal processes were overridden.

Findings concerning interactions

Software development projects do not operate in isolation, and interactions are needed in order to identify the stakeholders’ needs and requirements. Information and knowledge exchange and generation, co-design and co-creation are carried through interactions (Berthon & John 2006). As discussed in Section 7.4, the elements of interactions are communication, collaboration and integration. Chen (2007) considers communication and collaboration as enablers for successful interaction. Vijaykumar and Chakrabarti (2007) argue that communication is the key element of collaboration. Kahn (1996) suggests that cross-functional
integration consists of interaction and collaboration. Communication and coordination are seen as enablers for cross-functional software development by Botzenhard et al. (2011). In summary, the interaction elements can be considered as communication, collaboration and integration. Interactions are seen as successful when they are coordinated formally and collaboration reflects harmonious interdepartmental relationships (Atuahene-Gima 2005, Ruekert & Walker 1987).

Based on the empirical data, there are interactions between software productization and the major stakeholders. Communication is the major element of collaboration and relationships. It is needed to receive and understand stakeholders and their expectations, but also to synchronize cross-functional software product planning and development. Visibility is an important part of establishing good communication. It is provided through sharing plans, progress, and statuses and using common software systems and tools. Visibility is also dependent on willingness to share things proactively and mutual trust. Unwillingness to share feeds uncertainty and breaks confidence and trust, leaving room for rumors and false assumptions. Informants stressed the importance of good-quality documentation and up-to-date plans to avoid misunderstandings and rework. The use of common software systems was seen as beneficial for communication, but only if used in a disciplined way. Communication is supposed to be two-way in order to be effective. The findings revealed that the communication with most powerful and influential stakeholders such as marketing, corporate management and product and portfolio management was considered to be one-way only, and there was no formal forum to influence, negotiate or give feedback on the decisions or plans made.

Collaboration is needed to achieve common goals. Open and honest two-way communication was seen as the key element of collaboration. Informants highlighted all key stakeholders’ involvement in planning to promote collaboration in order to create successful software products. In the case company, the collaboration often relies on personal activity and willingness and capability to network, and is rarely coordinated by formal processes or practices. According to the informants, unclear roles and processes are major obstacles to collaboration. The processes are well-documented, trained on and easily available for personnel, but the company is in a transition phase of its operational mode, therefore some organizations are applying some old processes and some new ones. It was not always known which process should be applied. Integration is seen as a synonym for collaboration, with one distinguishing factor. Integration is embedded in
processes and practices, i.e. the milestone model which is the master tool for synchronizing all product development activities and deliveries. The milestone model defines the maturity of each delivery and its order, despite the different development methods used in different development teams.

Two groups were identified among the stakeholders. The ‘management group’ consists of stakeholders such as marketing, corporate management and product and portfolio management. The second stakeholder group can be called the ‘software development group’, involving software development professionals in software platform development and software productization. The findings indicated that the interaction between these two stakeholder groups was mainly one-way communication.

Summary

This section summarizes the differences between the a priori model and the a posteriori model. The a priori model was based on theory derived from the literature, and the a posteriori model was built upon the empirical findings.

Similarities

- Both models identify the same stakeholders.
- All stakeholders in the models receive and grasp market information.
- The value or product is delivered to the market and customers as an outcome of the product development.
- The a priori model is based on Weerd’s et al. (2006b) framework, in which the stakeholders of software product management are portfolio management, roadmapping, release planning and requirement management, and release planning.

Differences

- In the a posteriori model all identified stakeholders are relevant for software product management and productization.
- The a posteriori model derived from the empirical data shows how productization consists of two elements, namely from the conceptual model of software productization and software productization. The conceptual model is about the planning phase of the software product and the latter part is the implementation phase of the software product. The model does not take
into account the product development method used (for example sequential, iterative or spiral). The same stakeholders are involved with both the planning and implementation phases.

− In the a priori model the interactions consist of three elements: communication, collaboration and integration. In the a posteriori model interactions consist of communication and collaboration. Integration is seen as a synonym or part of collaboration.

− In the more detailed model of stakeholder interactions derived from the a posteriori model, the strong grouping of stakeholders is evident. The management group and the software development group seem to have very weak interactions between them, and the interactions are strong and focused mainly between the stakeholders in each group.

The differences reflect that the software product management and productization within are not as straightforward a set of actions carried out by a nominated stakeholder as stated in the literature. Instead there are several stakeholders who may influence, and want to influence in many ways the software product in the productization phase.

8.3 Managerial implications

The results show that the complexity of creating software products to the market arises from it being a multi-disciplinary process including many stakeholder aspirations and views. A company can increase software product development efficiency and reduce costs by rethinking the software productization phase. The main point is to ensure that the processes regarding the mobile software product content definition such as features and functionalities are clearly defined early enough, and accepted by all of the most important stakeholders.

Productization is a method for meeting customer and segment-specific needs more efficiently. The method supports a production environment in which the core components of the product and the customer-specific layer are identified and separated. This enables the configuration of new products from existing core components and the development team may focus only on the customer-specific part. It leads into an increased capability to produce customer-specific product variants more efficiently. The productization approach applied in software product development can be seen to address the challenges of:
More effective production of software
- Bringing new products more quickly and efficiently to the market
- Meeting customer and segment specific needs more efficiently
- Getting new products to the market on time.

As productization emphasizes the separation of the core components and the customer or segment-specific extension of the software product, it enables the development team to focus on the development of the extension part of each variant of the software product. Additionally the accelerated production of software and faster time to market may be gained through setting up proficient, visible and flexible decision making processes. It is achieved through the identification of stakeholder interactions and their influence. The integration of two stakeholder domains, namely the management group and the software development group, strongly enhances stakeholder interactions leading to increased visibility in software product related decision making.

**Identification of stakeholders and their influence**

Firstly, to be able to understand stakeholder interactions, the stakeholders should be identified. The first step is to create a physical map illustrating those stakeholders who have any interest in the software product. It is beneficial to draw the connections between software productization and stakeholders, but also to include in the map how stakeholders are connected to each other. Knowing the stakeholders and the connections between them is not sufficient without also identifying their level of interest. Their influence should be analyzed and described to understand the specific matter that makes the stakeholder interested in that particular software product. What are the drivers for each stakeholder? What kind of authority is the stakeholder using or able or willing to use? It is also important to understand who the stakeholders are who can progress or hinder the software project. This map help the software project manager to understand the environment in which they operate, and also to consider stakeholders early even if they do not interact actively in the software productization phase but perhaps might do so later. The results suggest that managers involved with software productization adopt stakeholder identification as a systematic procedure in all software productization projects.

Secondly, clearly defined processes with roles and responsibilities included advance interactions. This may sound apparent, but in a large cross-functional
product development company, common understanding about roles and responsibilities may be unclear, even though the processes are professionally documented, trained on, and available. In the case company, multiple processes were applied simultaneously due to the on-going large operational mode transformation. The data showed that it was difficult to understand the chain of command and responsibilities of the roles outside software productization. Creating a map of stakeholders and connections using real names may prevent this confusion and ensure that the software development team knows whom to contact.

**Identification of interactions between stakeholders**

In the context of cross-functional software development, all elements of interactions are needed between software productization and all key stakeholders in order to create successful software products.

Communication transmitting stakeholders’ needs and requirements is not sufficient: collaboration and integration are needed, too. Providing visibility of plans is not enough, but discussion and common planning meetings are also needed. Informants preferred face-to-face meetings but it is more practical to have meetings via teleconference and net meetings. In these meetings, different views are discussed and decisions are made. From product development project perspective, these are also the forums in which the priorities of competing needs and requirements are agreed. In addition to decision-making, it is a forum for sharing cross-functional information, problems faced and possible delays in certain technology areas. The common view of the product is ensured and strengthened. One way to ensure two-way communication is to establish regular formal and common planning and review meetings in which all stakeholders have representatives and the monthly updates and priorities of the roadmap and master backlog are discussed together. This ensures an opportunity to provide feedback, further questions and more detailed justifications of needs and requirements openly. A clearly defined communication process calms turbulent software productization and reduces direct and overlapping interfaces with multiple stakeholders, leaving space to focus on implementation work rather than putting effort into competing demands and stakeholders’ personal ambitions. It also reduces the risk for human error. When the communication process is solid, it creates an environment in which mobile software product development teams can rely on obtaining requirements through formal decision-making and allocating
resources to the appropriate tasks. A formal communication process postulates clearly defined roles and responsibilities.

Collaboration is needed to ensure that all stakeholders and development co-teams are sharing and working towards the same target. As mobile software products are very complex, with numerous dependencies, it is essential to synchronize plans to optimize the work and minimize rework, idle time, and waiting. A disciplined way of using software systems supports both communication and collaboration. Two-way communication supports collaboration. Even when stakeholders in a management group seem to be remote and communication to be informative by nature, software project managers should be encouraged to invite and require stakeholders’ representatives to attend software release plan reviews and discussions. Two-way communication allows discussion with instant feedback and further questions for all stakeholders. If communication relies only on visibility of documentation and/or software systems, this leaves room for misunderstandings and lack of adequate information, and blocks discussion by the originator of the need or requirement as well as feedback and further questions. Collaboration is needed to gain deeper understanding about needs, requirements and prioritization. Discussions are needed for mutual feedback and sharing views, but also for bringing new perspectives to the table to be considered. Integration is enabled by documented processes in which the interfaces between stakeholders are formally defined.

**Emphasizing visibility and discipline**

Visibility of decision-making and justifications is crucial for obtaining a confident working atmosphere. This concerns product-related decisions such as prioritization of products and individual needs and requirements, plans and progress. Visibility of decision-making can be enabled by software systems and the disciplined use of processes – by all stakeholders.

It is difficult to manage software needs and requirements when they come via various routes. This means that formal prioritization mechanisms cannot be applied. If the change information does not reach all functions involved in product development, valuable time and resources are wasted. In the case organization, much attention has been paid to improving requirement and change management processes. The process is supported by an information system and the tool is the primary method for communicating and providing visibility, and ensuring that all implementation teams have access to product changes and the
latest software needs and priorities. Having one information system as a master system ensures that software implementation teams are able to plan their daily work and can be confident that they are maximizing their effort in doing the right things. It is suggested that all new needs and requirements from stakeholders which bypass formal processes are subjugated to formal processes to ensure their visibility to decision-making and justifications of new needs. Needs and requirements received any other way are ignored and no exceptions are accepted.

Stakeholders who override processes and formal decision-making and pass new requirements and needs via emails and phone calls should not be considered by software productization. To prevent this happening, stakeholders could be trained to apply the processes the consequences of informal processes should be highlighted to them. It is crucial that all stakeholders apply the defined processes and do not override them. A proposal could be that major new software requirements and needs which require reverse engineering in software platform development, and stakeholders who override processes, are not considered in software productization, instead stakeholders are trained and guided to transmit the requirement through formal processes and formal decision-making.

The findings indicated that major software requirements which are received in late phase of software productization increase the risk of software product delays. Even though the role of software productization is to gather new software requirements, major ones causing changes in the software architecture of the software platform should be avoided. Major software changes at a late phase of software product development affecting fundamental early phase decisions or assumptions should be avoided. Late phase requirements increase the risk of delayed product launches. Features that have been added hastily increase the number of errors, leading to an increased amount of software testing and error correction work. Late phase requirements should also be subjugated under formal processes and decision-making procedures.

In summary, the findings confirm that the intensity of interactions between software productization and stakeholders needs to be improved. Few important actions could be considered to calm software productization, which is working under enormous pressure to create the mobile software product on time. According to the empirical data marketing, corporate management and product and portfolio management play a key role in clarifying the role and tasks of software productization. This could be done mainly through paying attention to and sharpening a disciplined way of working, training and guiding stakeholders and making the impacts visible, for example the cost of rework, delays in
deliveries, number of errors and testing effort needed. Because of the stakeholders’ managerial position and authority, this should be done with discretion, but making impacts visible and transferring them into financial figures increases everyone’s understanding and justifies the need for formal procedures even though they might be perceived as bureaucratic. The adoption of all elements of interactions will increase cross-functional connectedness and extenuate misunderstanding and conflicts between stakeholders.

8.4 Limitations of the research

There are limitations in this research. The first limitation relates to the generalizability of the results. The study was conducted in a large international company. Due to the fact that only one company was analyzed, it cannot be said whether the results can be applied to all large companies in general or can be expected to be relevant in similar contexts. Further studies that replicate this research could increase confidence in the results. Based on the empirical data, this study is able to explain how stakeholder interactions appear in cross-functional productization in a company.

The second strong limitation is the sample of interviewees. The interviewees represent developers involved with productization, and representatives from the management group are not included. The results revealed a lack of interaction between management-oriented and development-oriented groups. The interviewees were representatives from the development-oriented group, so the other view was not investigated in this research. The purpose of this study was to focus on how the software development community perceives the productization, and that can be explained in this research.

The third limitation is the risk of researcher’s retrospective bias caused by participant observation. The researcher is not able to work as an external observer, and this may increase the risk of manipulation and participation instead of observation. This was a private project of the researcher, and a research group would have been beneficial in order to have peers to discuss the analysis phase. However, the supervisors of the research compensated for the lack of a research group. It was also possible to have discussions with interviewees during the study to ensure that the empirical data was understood correctly.
8.5 Suggestions for future research

This study provides a view of stakeholder interactions in the context of cross-functional software productization and as such constitutes a contribution to the stakeholder interaction literature. Stakeholder interactions in cross-functional software product development are an interesting topic, and this study leaves many questions unanswered. The organizational context provides numerous interesting questions. As one study can only focus on a few questions, future research raised by the results of this study from new perspectives is needed.

Ideas for future research arise from the choices of this study. Only one stakeholder’s perspective was explored, due to the scope of this research. Therefore, it would be interesting and useful to study interactions from other stakeholders’ perspectives, for example how corporate management or product and portfolio management perceive the interactions between them and software productization. This research focused on middle managers: the framework could be used for studying the viewpoint of software practitioners.

The results of this study raise a need to study the most intensive interactions identified more deeply. Furthermore, it would be interesting to widen the scope of the stakeholders of software productization. While this thesis is focused on interactions of intra-organizational stakeholders, another area for future research could include external stakeholders such as end users, consumers and partners or subcontractors. Their role in large-scale software product development is critical. Additionally, it would be interesting to study and understand more deeply the most intensive interactions identified in this study.

Another starting point for future research would be focusing on the interactions of unsuccessful software productization. The empirical project data for this research consisted of more (or less) successful projects, but no projects which were unsuccessful and stopped. The world might be seen very differently from an unsuccessful project perspective.
9 Conclusions

This chapter concludes the research. Next, the research questions are answered in respect to the empirical findings and the theoretical method applied, and this is followed by some concluding words.

9.1 Answers to the research questions

The focus of this study was: Stakeholder interactions in cross-functional software productization. Software productization is the phase in which the software product is finalized for launching to the market. It can be considered as a software development project within a product development project. The trigger for software productization is an improved capability to differentiate software products, to achieve faster time to market and a capability to respond faster to market or customer needs and business opportunities. During software productization, a customer- or product-specific layer is added on top of the core software product. The final outcome of software productization is an extended software product. Software productization is based on software platform delivery consisting of core software components, meaning that the basic software technology, enablers and functionalities are established in the software platform. No new major software requirements impacting on the software architecture or other fundamental choices regarding the software technology should be implemented in software productization, because this increases the risk of delayed software products by jeopardizing the launch hitting the market window.

What stakeholder interactions exist in cross-functional software productization?

The first research question was: What stakeholder interactions exist in cross-functional software productization? The success of software product creation relies heavily on proficient interactions between stakeholders. Many stakeholders representing different organizations and functions of the company are involved with software productization. Stakeholders have an important role in the success of software productization, i.e. being able to deliver the finalized software product to the market window in time and as planned. In order to be able to analyze the interactions, the influential stakeholders must be identified first. Analysis of the data revealed that internal stakeholders of software productization were vaguely
identified and no formal process of stakeholder identification or mapping or evaluation of stakeholders’ power and interest existed or were used. The most important stakeholders which have most of the power and authority over software productization are marketing, corporate management and product and portfolio management. Software platform development was recognized as a stakeholder, too, because software products are developed on top of the software technology enabled by software platform development. Because software productization can be considered as a software project within a product development project, product development is not seen as a separate stakeholder from a software productization perspective but rather as a home base for it.

Successful software product creation is enabled by common understanding about the product. What should it be like? To whom it is targeted? Stakeholders share this view and influence software productization through interactions. It is crucial that the interactions are synergistic, consultative and discursive by nature to optimize the outcome of the software productization. The elements of interaction are communication, collaboration and integration. These elements are used diversely and the intensity of the interactions varies among stakeholders. The research showed that there were two groups of stakeholders. The management group consisted of marketing, corporate management and product and portfolio management: they interact with each other intensively using all elements of interaction. The software development group consists of software platform development and software productization, in which software developers and managers of software development organizations interact intensively, likewise using all elements of interaction.

How do stakeholders associate in cross-functional software productization?

The second research question was: How do stakeholders associate in cross-functional software productization? The purpose of this research question was to examine how interactions are carried out between software productization and its stakeholders. Interactions consist of communication, collaboration and integration and are needed to exchange information and knowledge, and to explore and identify stakeholders’ needs and expectations. Communication is acknowledged as the most important element of interaction within software productization, and it turned out to be the only element of interaction used between the most influential stakeholders and software productization.
The evidence showed how stakeholders are divided into two groups. The interaction between these two groups was mainly carried out through communication and it was interesting to see that the communication was seen by the software development group to come from the top down and from management to software development communities, considered to be one-way only due to a lack of formal channels for mutual discussion and the ability to respond, question, present more detailed questions and give instant feedback from software productization. Communication is conducted mainly through documents such as the product roadmap and master backlog, which are delivered monthly via email, including a link to the internal company intranet pages or Wikis with restricted visibility. Occasionally direct phone calls and emails are received from stakeholders in the management group expressing new needs and requirements and bypassing the formal processes. One-way communication blocks mutual information exchange and increases the risk of misunderstandings and delays in software product launches. Needs and requirements received beyond formal processes increase uncertainty and prevent visibility of decision-making and justifications of new needs.

Integration is identified between marketing and software productization in practice through a product manager that is allocated to the product development project as a representative of marketing. In that role, the product manager is focused on being a voice of the product-specific market and customer and preparing the go-to-market and product launch. Collaboration is considered in terms of common planning and review meetings, workshops and mutual discussions. Collaboration is seen as missing between software productization and the stakeholders of management group.

Due to the fact that software productization can be considered as a software project within a product development project, intensive interaction between the product development project and software productization is natural. All elements of interactions are in use. Communication starts in practice as soon as the product development project is established. The visibility of documentation, plans and progress and common meetings establish a good basis for two-way communication. Visibility is also often supported by software systems. Integration can be considered as consisting of the same elements as collaboration, but additionally it is ensured through formal processes such as the milestone model which is applied in product development and in which all different organizations and technology areas are synchronized as one master plan. Integration is built-in in processes and in the milestone model applied. In order to
create a successful product release for the product development project, intensive collaboration is needed. Collaboration is carried out through common planning meetings and workshops. The success of software productization depends on the maturity of the Product Release delivered by the software development platform.

The major concern from a software productization perspective is that the interaction between the most influential stakeholders is based on one-way communication through documentation: the product roadmap and master backlog. According to the interviews, there is no formal procedure to negotiate, give feedback, make further questions or discuss the product roadmap or master backlog with corporate management, marketing or product and portfolio management. Their role, however, is crucial for successful software products.

**How could stakeholder interactions be coordinated in cross-functional software productization?**

The third research question was: *How could stakeholder interactions be coordinated in cross-functional software productization?* The ability to clearly understand and consider stakeholders’ needs and expectations is a critical element for company success. A key factor for supporting stakeholder interactions is to create an interaction process or protocol to ensure that all the elements of interactions are required and supported. A defined interaction process could promote and support interactions, for example regular common planning and reviews of the product roadmap and master backlog. Listening to worries or comments about the roadmap or master backlog is not enough, but a change in attitude is a prerequisite for shared and mutual understanding. The stakeholders must be identified and documented and their influence, interest and power on corporate, people, resources and software products estimated systematically in software productization. This can be done, for example, by writing down their drivers or interests for the software product at hand, and their position in the company. It is important to understand what kind of power they have regarding people, resources and funding for the software product. Furthermore, it is beneficial to explore and illustrate stakeholders’ connections, e.g. interactions with software productization, but also how stakeholders interact with each other, and how the interactions are carried out. Is the interaction based on communication, or are there some practices and processes in place which establish collaboration and integration? The organizational structure of a large company is complex, and sometimes it may be difficult to identify the exact
person to contact. Therefore stakeholders should be identified at name level and ensure that the names given are truly representing the stakeholder.

The evidence showed that software development communities apply strictly defined processes. This should be required from all stakeholders. One major software requirement received informally from an influential stakeholder during the software productization phase may risk the software product launch. Software productization must be strict with its actions: no major new needs and requirements not received through formal processes should be accepted. The consequences of overridden software release and requirement management process and late phase new major software requirements during software productization should made visible, for example the impact on the cost and schedule of the software product.

### 9.2 Concluding words

This research has studied stakeholder interactions in cross-functional software productization. The target of this research is to increase awareness of the importance of interactions in cross-functional software productization. Hopefully this study provides some help for software productization in considering and outlining the surrounding interactions, making it easier to designate the consequences to stakeholders.

At the completion of this research, as a researcher, the author has gained much experience and knowledge from this journey. There were two reasons why the author became interested in starting this thesis. Daily work in a research and development organization maintains the hunger for studying and learning. The daily work also raised some questions to which the author wanted to find answers and which they wanted to understand more deeply. This research revealed that the complexity of interactions is even higher than the author expected, and many questions are still unanswered. Software productization research is still a fairly unexplored field, and a large number of aspects were not covered in this research. However, I hope that other researchers may find this study interesting and enhance the research field in the future.
References


Battin R & Crocker R & Kreidler J & Subramanian K (2001) Leveraging resources in
improvement in software product management. Thesis report: INF/SCR-08-09. The
Netherlands, University Utrecht.
Berthon P & Joby J (2006) From entities to interfaces: Delineating value in customer-firm
interactions. In: Lusch R & Vargo S (eds) The service-dominant logic of marketing,
66.
Boddy D & Buchanan D (1999) Take the lead: interpersonal skills for project managers.
New York, Prentice Hall.
28(2): 1–12.
York, John Wiley & Sons Inc.
explanation for the persistence of high velocity competition. Organization Science
Boh W & Ren Y & Kiesler S & Bussjaeger R (2007) Expertise and collaboration in the
Bond E & Houston M (2003) Barriers to matching new technologies and market
120–135.
development projects and project performance. The Journal of Product Innovation
Bosch J (2000) Design and use of software architectures: adopting and evolving a product-
line approach. Harlow, Addison-Wesley.
(ICSE 2001): 91 –100.
Bosch J (2002) Maturity and evolution in software product lines: approaches, artifacts and


Cooper R (1993) Winning at new products: accelerating the process from idea to launch. 2nd Ed. Addison Wesley.


249–258.
leadership. New York, NY, Free Press.
Miller R & Olleros X (2001) Project shaping as a competitive advantage. In: Miller R &
Lessard DR (eds) The strategic management of large engineering projects: Shaping
Minderhoud S & Fraser P (2005) Shifting paradigms of product development in fast and
Mishra D & Mishra A & Ostrovska S (2012) Impact of physical ambiance on
communication, collaboration and coordination in agile software development: An
Mitchell R & Agle B & Wood D (1997) Toward a theory of stakeholder identification and
salience: Defining the principle of who and what really counts. Academy of
mobile-marketing-tools/latest-mobile-stats/a#subscribers/. Cited 2014/01/12.
Moenaert R & Souder W (1990) An information transfer model for integrating marketing
and R&D personnel in new product development projects. Journal of Product
Moenaert R & Souder W & De Meyer A & Deschoolmeester D (1994) R&D-marketing
integration mechanisms, communication flows, and innovation success. Journal of
Molin-Juustila T (2006) Cross-functional interaction during the early phases of user
centered software new product development: Reconsidering the common area of
interest. Dissertation. University of Oulu, Faculty of Science, Department of
Information Processing Science.


198


Salo O (2006) Enabling software process improvement in agile software development teams and organisations. Helsinki, VTT.


Sipilä J (1996) Asiantuntijapalveluiden tuotteistaminen. WSOY.


Appendix 1 List of interviews conducted for the study

Interview method F2F corresponds to interviews conducted in a face-to-face meeting. Net corresponds to interviews conducted via an Internet meeting (teleconference and video used).

Table 11. List of interviews conducted for the study.

<table>
<thead>
<tr>
<th>Code</th>
<th>Informant’s role</th>
<th>Company experience in years</th>
<th>Interview date</th>
<th>Time</th>
<th>Interview method</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Software organization manager</td>
<td>20</td>
<td>10:00 – 11:30</td>
<td>F2F</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>I2</td>
<td>Product development manager</td>
<td>22</td>
<td>09:00 – 10:30</td>
<td>F2F</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>I3</td>
<td>Software error manager</td>
<td>12</td>
<td>12:30 – 14:00</td>
<td>F2F</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>I4</td>
<td>Software error manager</td>
<td>16</td>
<td>15:30 – 17:00</td>
<td>F2F</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>I5</td>
<td>Product development manager</td>
<td>21</td>
<td>12:30 – 14:00</td>
<td>F2F</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>I6</td>
<td>Technology manager</td>
<td>16</td>
<td>15:00 – 16:30</td>
<td>Net</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>I7</td>
<td>Software requirement manager</td>
<td>11</td>
<td>09:30 – 11:00</td>
<td>F2F</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>I8</td>
<td>Technology roadmapping manager</td>
<td>13</td>
<td>12:00 – 14:30</td>
<td>F2F</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>I9</td>
<td>Architecture team leader</td>
<td>18</td>
<td>09:30 – 11:00</td>
<td>Net</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>I10</td>
<td>Technology roadmapper</td>
<td>14</td>
<td>13:30 – 15:00</td>
<td>Net</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>I11</td>
<td>Team leader technology planning</td>
<td>17</td>
<td>10:00 – 11:30</td>
<td>F2F</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>I12</td>
<td>Chief architect</td>
<td>15</td>
<td>15:00 – 16:30</td>
<td>Net</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>I13</td>
<td>Product development manager</td>
<td>10</td>
<td>10:00 – 11:30</td>
<td>Net</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>I14</td>
<td>Product development manager</td>
<td>11</td>
<td>12:30 – 14:00</td>
<td>Net</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>I15</td>
<td>Technology manager</td>
<td>19</td>
<td>15:00 – 16:30</td>
<td>F2F</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>I16</td>
<td>Software project manager</td>
<td>20</td>
<td>09:00 – 10:30</td>
<td>F2F</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>I17</td>
<td>Software project manager</td>
<td>12</td>
<td>12:00 – 13:30</td>
<td>F2F</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>I18</td>
<td>Chief architect</td>
<td>20</td>
<td>11:30 – 13:00</td>
<td>F2F</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>I19</td>
<td>Software organization manager</td>
<td>12</td>
<td>11:30 – 13:00</td>
<td>Net</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>I20</td>
<td>Organization manager</td>
<td>15</td>
<td>14:00 – 15:30</td>
<td>F2F</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>314</td>
<td></td>
<td>118</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>15.7 years</td>
<td>1.5 hours</td>
<td>5.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix 2 Interview guide

The following interview guide served to orient the conversation during the interview.

1. What is your role and responsibilities?
2. How are product programs/projects started/initiated?
3. Where do product program/project targets and priorities come from?
4. Who are your stakeholders?
5. How is product program planning done?
6. How is progress measured and followed up?
7. How are risks managed (what kind of risks, level of risks) during product development?
8. What are the barriers for cross-functional collaboration? How are the barriers addressed?
9. What are the challenges for successful product release (PR1)?
10. Is there anything else you would like to add? Did we miss anything in our talk?
<table>
<thead>
<tr>
<th>Code</th>
<th>Document</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>Tools landscape presentation 2011</td>
<td>11</td>
</tr>
<tr>
<td>N2</td>
<td>Milestone framework 2011</td>
<td>13</td>
</tr>
<tr>
<td>N3</td>
<td>Requirement management process 2010</td>
<td>11</td>
</tr>
<tr>
<td>N4</td>
<td>Strategy planning process 2009</td>
<td>23</td>
</tr>
<tr>
<td>N5</td>
<td>Roadmap model 2009</td>
<td>1</td>
</tr>
<tr>
<td>N6</td>
<td>Marketing integration process 2008</td>
<td>27</td>
</tr>
<tr>
<td>N7</td>
<td>Software blueprint 2008</td>
<td>4</td>
</tr>
<tr>
<td>N8</td>
<td>Platform and asset creation process 2008</td>
<td>28</td>
</tr>
<tr>
<td>N9</td>
<td>Agile software engineering process 2008</td>
<td>30</td>
</tr>
<tr>
<td>N10</td>
<td>Productization process 2008</td>
<td>52</td>
</tr>
<tr>
<td>N11</td>
<td>Product and portfolio management process 2008</td>
<td>41</td>
</tr>
<tr>
<td>N12</td>
<td>Product creation process 2008</td>
<td>25</td>
</tr>
<tr>
<td>N13</td>
<td>Reasoning timebox presentation 2007</td>
<td>1</td>
</tr>
<tr>
<td>N14</td>
<td>Experience development train planning training 2010</td>
<td>72</td>
</tr>
<tr>
<td>N15</td>
<td>Current state analysis summary 2010</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Sum of pages</td>
<td>347</td>
</tr>
<tr>
<td>P1</td>
<td>Project 1 report 2010</td>
<td>26</td>
</tr>
<tr>
<td>P2</td>
<td>Project 2 report 2007</td>
<td>3</td>
</tr>
<tr>
<td>P3</td>
<td>Project 3 report 2008</td>
<td>1</td>
</tr>
<tr>
<td>P4</td>
<td>Project 4 report 2009</td>
<td>27</td>
</tr>
<tr>
<td>P5</td>
<td>Project 5 report 2009</td>
<td>33</td>
</tr>
<tr>
<td>P6</td>
<td>Project 6 report 2010</td>
<td>5</td>
</tr>
<tr>
<td>P7</td>
<td>Project 7 report 2008</td>
<td>4</td>
</tr>
<tr>
<td>P8</td>
<td>Project 8 report 2010</td>
<td>9</td>
</tr>
<tr>
<td>P9</td>
<td>Project 9 report 2011</td>
<td>9</td>
</tr>
<tr>
<td>P10</td>
<td>Project 10 report 2010</td>
<td>30</td>
</tr>
<tr>
<td>P11</td>
<td>Project 11 report 2010</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>Sum of pages</td>
<td>191</td>
</tr>
<tr>
<td>D1</td>
<td>Author’s personal work diary 1</td>
<td>214</td>
</tr>
<tr>
<td>D2</td>
<td>Author’s personal work diary 2</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>Sum of pages</td>
<td>294</td>
</tr>
<tr>
<td></td>
<td>Total number of pages</td>
<td>832</td>
</tr>
</tbody>
</table>
## Appendix 4 Coding used in the analysis in NVivo

<table>
<thead>
<tr>
<th>Theme</th>
<th>Category</th>
<th>Sub category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product development</td>
<td>Capacity management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Collaboration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Communication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Decision making</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Early involvement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Good practices</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improvements needed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Portfolio management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prioritization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Processes roles</td>
<td>Agile and lean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multiple milestone models</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Process and culture</td>
</tr>
<tr>
<td>Product dev phases</td>
<td>Capacity and resources</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Challenges in PR1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Content definition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cooperation and barriers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSA project planning</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Late or missing requirements and needs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project follow-up</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project planning</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project start and initiation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project targets and priorities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stakeholder identification</td>
<td></td>
</tr>
<tr>
<td>Stakeholders</td>
<td>Corporate management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marketing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Platform development</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Portfolio management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Product development</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rel planning and req management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Roadmapping</td>
<td></td>
</tr>
<tr>
<td>Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visibility</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
612. Lawrence, Carl (2013) Innovating with information technology in a globalized world: being proactive about culture
613. Ardanov, Pavlo (2013) Priming capacities of endophytic Methylobacterium sp. on potato (Solanum tuberosum L.)
615. Holm, Jana (2013) Catalytic pretreatment and hydrolysis of fibre sludge into reducing sugars
616. Kemi, Ulla (2013) Adaptation to growing season length in the perennial Arabidopsis lyrata
618. Rodríguez, Pilar (2013) Combining lean thinking and agile software development: how do software-intensive companies use them in practice?
619. Vatka, Emma (2014) Boreal populations facing climatic and habitat changes
620. Isomursu, Marja (2014) Host–parasite interactions of boreal forest grouse and their intestinal helminth parasites
622. Matusek, Florian (2014) Selective privacy protection for video surveillance
623. Virtanen, Elina (2014) Effects of haulm killing and gibberellic acid on seed potato (Solanum tuberosum L.) and techniques for micro- and minituber production in northern latitudes
624. Kopatz, Alexander (2014) Genetic structure of the brown bears (Ursus arctos) in Northern Europe
625. Loukola, Olli (2014) Information networks among species: adaptations and counter-adaptations in acquiring and hiding information
626. Langrial, Sitwat (2014) Exploring the influence of persuasive reminders and virtual rehearsal on the efficacy of health behavior change support system
Heli Päätalo

STAKEHOLDER INTERACTIONS IN CROSS-FUNCTIONAL PRODUCTIZATION

THE CASE OF MOBILE SOFTWARE DEVELOPMENT