Department of Computer Science

Linking Long-Term Product Planning and Requirements Prioritization to Customer Value Creation

Laura Lehtola-Karttunen

Aalto University

DOCTORAL
DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 217/2015

Linking Long-Term Product Planning and Requirements Prioritization to Customer Value Creation

Laura Lehtola-Karttunen

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall T2 of the school on 29 January 2016 at 12.

Aalto University
School of Science
Department of Computer Science
Preago Research Group

Supervising professor
Dr. Marjo Kauppinen

Thesis advisor
Dr. Tomi Männistö

Preliminary examiners
Professor Barbara Paech, University of Heidelberg, Germany
Assistant Professor Maya Daneva, University of Twente, Netherlands

Opponent
Professor Mike Mannion, Glasgow Caledonian University, Scotland

Aalto University publication series
DOCTORAL DISSERTATIONS 217/2015

ISBN 978-952-60-6590-8 (printed)
ISBN 978-952-60-6591-5 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-6591-5

Unigrafia Oy
Helsinki 2015

Finland

Publication orders (printed book):
Laura.Lehtola@iki.fi

NORDIC ECOLABEL
441 697
Printed matter

Aalto University

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Laura Lehtola-Karttunen

Name of the doctoral dissertation
Linking Long-Term Product Planning and Requirements Prioritization to Customer Value Creation

Publisher School of Science

Unit Department of Computer Science and Engineering

Series Aalto University publication series DOCTORAL DISSERTATIONS 217/2015

Field of research Software Engineering

Manuscript submitted 9 September 2014 **Date of the defence** 29 January 2016

Permission to publish granted (date) 9 September 2015 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

Creating value for different customer segments is essential to the business of a company. Thus, software product development companies' ability to implement the most valuable requirements in their products has been seen as critical. The literature offers requirements prioritization methods for selecting requirements, but their suitability for solving practical challenges is not clear. The state of the practice in long-term product planning and requirements prioritization, and the practical challenges involved is not thoroughly analysed. Therefore, the connection between the selection of product features and customer value creation is also an area that needs more investigation.

This thesis investigates the current state of long-term product planning and requirements prioritization, and their linkages to customer value creation in market-driven software product development. The results are based on the experience gathered from 7 Finnish software product development companies that had recognized the importance to improve their long-term product planning and requirements prioritization practices. To gain a deep understanding in real product development context, we conducted longitudinal case studies using an action research approach. In addition, we evaluated the suitability of requirements prioritization methods in product development projects.

The thesis provides a systematic analysis of long-term product planning and requirements prioritization activities and challenges involved in the market-driven software product development companies. According to our results product planning activities are at a low-level, focusing on the individual features of individual products in short term. Practitioners have challenges in drawing different viewpoints together into product planning. Both integrating business and customer viewpoints to the project level as well as integrating the understanding about technical limitations and possibilities to product level planning was difficult in practice. The requirements prioritization methods did not solve the practical challenges.

On the basis of our findings we also propose a set of practices that support the link from long-term product planning and requirements prioritization to customer value creation. The essence of the suggested practices is 1) in the shift of planning focus from individual product features towards understanding the processes of customers and users and in describing the solutions from the customer viewpoint as a whole and 2) in the cross-functional effort to be able to combine the three main viewpoints of customer value creation (company's own business, customers & users, and implementation).

Keywords requirements prioritization, long-term productplanning,roadmapping,solution planning, market-driven requirements engineering, customer value, value creation

ISBN (printed) 978-952-60-6590-8 **ISBN (pdf)** 978-952-60-6591-5

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2015

Pages 175 **urn** http://urn.fi/URN:ISBN:978-952-60-6591-5

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Laura Lehtola-Karttunen

Väitöskirjan nimi
Linking Long-Term Product Planning and Requirements Prioritization to Customer Value Creation

Julkaisija Perustieteiden korkeakoulu

Yksikkö Department of Computer Science and Engineering

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 217/2015

Tutkimusala Software Engineering

Käsikirjoituksen pvm 09.09.2014 **Väitöspäivä** 29.01.2016

Julkaisuluvan myöntämispäivä 09.09.2015 **Kieli** Englanti

□ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Arvon luominen eri asiakassegmenteille on elintärkeää yrityksen liiketoiminnalle. Ohjelmistotuoteyrityksen kyvykkyys toteuttaa tuotteisiinsa asiakkaiden näkökulmasta katsoen arvokkaimmat ominaisuudet onkin siksi nähty kriittisenä. Kirjallisuudessa esitellään erilaisia priorisointimenetelmiä tuoteominaisuuksien valintaan, mutta niiden soveltuvuudesta käytännön ohjelmistotyöhön ei juuri tiedetä. Myöskään pitkäntähtäimen suunnittelun ja vaatimusten priorisoinnin nykytilaa sekä niihin liittyviä käytännön haasteita ei ole syvällisesti analysoitu. Siksi yhteyttä ominaisuusvalintojen ja asiakasarvon luomisen välillä onkin tärkeää tutkia lisää.

Tässä tutkimuksessa selvitettiin, miten pitkäntähtäimen tuotesuunnittelu ja vaatimusten priorisointi tukevat arvon luontia asiakkaille ja käyttäjille markkinaohjautuvassa ohjelmistotuotekehityksessä. Tulokset perustuvat seitsemästä suomalaisesta ohjelmistotuoteyrityksestä kerättyihin kokemuksiin. Saadaksemme tuotesuunnittelusta ja ominaisuusvalintojen tekemisestä niin pitkällä kuin lyhyelläkin tähtäimellä syvällisen ymmärryksen, toteutimme pitkittäisiä tapaustutkimuksia käyttäen toimintatutkimuslähestymistapaa. Lisäksi kokeilimme vaatimusten priorisointimenetelmiä todellisessa tuotekehitysympäristössä.

Tutkimus esittelee systemaattisen nykytila-analyysin ohjelmistoyritysten tuotesuunnittelun käytännöistä ja haasteista. Tulokset paljastavat, että ohjelmistotuotteita suunnitellaan pirstaleisesti, lyhyellä aikajänteellä ja yksittäisen tuotteen ominaisuuksiin kerralla keskittyen. Yritysten työntekijöillä on vaikeuksia käytännössä yhdistää arvonluonnin kannalta tärkeitä näkökulmia. Sekä liiketoiminta- ja asiakasnäkökulman tuominen tuotekehitysprojektien päätöksentekoon että toisaalta teknisten rajoitteiden ja mahdollisuuksien huomioiminen jo tuotesuunnittelun aiemmissa vaiheissa ovat erityisen haastavia.

Tutkimuksen tuloksena syntyi käytäntöjä, joiden avulla asiakasarvon luomista voisi tukea tuotesuunnittelussa. Näiden käytäntöjen keskiössä ovat 1) suunnittelun fokuksen siirtäminen yksittäisistä tuoteominaisuuksista asiakkaan ja loppukäyttäjän prosessien ymmärtämiseen, sekä ratkaisujen kuvaaminen asiakkaan näkökulmasta kokonaisuuksina, sekä 2) yli organisaatiorajojen ulottuva työskentely, jonka avulla voidaan paremmin yhdistää kolme tärkeintä asiakasarvon luonnin näkökulmaa (yrityksen oma liiketoiminta, asiakkaat ja käyttäjät, sekä tuotteen toteuttaminen).

Avainsanat vaatimusten priorisointi, pitkäntähtäimen tuotesuunnittelu, roadmapping, ratkaisusuunnittelu, markkinaohjautuva vaatimusmäärittely, asiakasarvo, arvon luominen

ISBN (painettu) 978-952-60-6590-8 **ISBN (pdf)** 978-952-60-6591-5

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2015

Sivumäärä 175 **urn** http://urn.fi/URN:ISBN:978-952-60-6591-5

Acknowledgements

Writing a dissertation is like writing a novel. Both are romantically referred as lonely journeys, but in reality neither could be done without intelligent and clever fellow travellers.

Without any overstatement: This dissertation would not exist without Professor Marjo Kauppinen. As a supervisor, she is at the same time inexplicably demanding but still flexible. Thank you for so much. I would also like to thank my other supervisor Professor Tomi Männistö for his comments and support during the years.

The articles in this thesis are more or less joint efforts. I would like to warmly thank all the co-authors of the papers, Sari Kujala, Jarno Vähäniitty, Marko Komssi, Alan Davis, Harri Töhönen, and Marjo Kauppinen. Thank you. It has been pleasure to work with all of you.

This work also contains significant influences from many other inspiring people at Aalto University that have shared their views and knowledge with me. Timo Asikainen, Danielle Pichlis and Stefanie Hofemann, thank you for preparing the pictures of the dissertation and articles.

The role of the case companies and the individual practitioners within the companies in this kind of research is huge. My warmest thanks are dedicated to you all.

This research was financially supported by TEKES (the Finnish Funding Agency for Innovation) and a graduate assistantship of the Helsinki Graduate School in Computer Science and Engineering (HeCSE). I am grateful for this support.

The preliminary examination of this dissertation was done by Professor Barbara Paech and Assistant Professor Maya Daneva. Their comments and valuable feedback improved this dissertation and prepared me for the final defence.

Finally, I want to thank my family. Not that much for anything related to thesis, but for everything else. Tuomas, Joonatan and Matilda, thank you for being who you are.

Helsinki, 1 December 2016
Laura Lehtola-Karttunen

Contents

List of Publications

This doctoral dissertation consists of a summary and of the following publications, which are referred to in the text by their roman numerals.

I. LEHTOLA, L., KAUPPINEN, M. and KUJALA, S. (2004) Requirement Prioritization Challenges in Practice, *5th International Conference on Product Focused Software Process Improvement*, 497-508.

II. LEHTOLA, L. and KAUPPINEN, M. (2006) Suitability of requirements prioritization methods for market-driven software product development, *Software Process: Improvement and Practice* (11:1), 7-19.

III. LEHTOLA, L., KAUPPINEN, M. and KUJALA, S. (2005) Linking the Business View to Requirements Engineering: Long-Term Product Planning by Roadmapping, *13th IEEE International Requirements Engineering Conference (RE'05)*, 439-443.

IV. LEHTOLA, L., KAUPPINEN, M., VÄHÄNIITTY, J. and KOMSSI, M. (2009), Linking Business and Requirements Engineering: Is Solution Planning a Missing Activity in Software Product Companies? *Requirements Engineering (*14:2), 113-128.

V. KAUPPINEN, M., J. SAVOLAINEN, L. LEHTOLA, M. KOMSSI, H. TÖHÖNEN and A. DAVIS. (2009) From Feature Development to Customer Value Creation, *17th IEEE International Requirements Engineering Conference (RE '09)*, 275 – 280.

Author’s Contribution

Publication I: Requirement Prioritization Challenges in Practice

Publication II: Suitability of Requirements Prioritization Methods for Market-Driven Software Product Developmen

Publication III: Linking the Business View to Requirements Engineering: Long-Term Product Planning by Roadmapping

The author of this thesis is the principal author of the publications 1-3. She is responsible for conducting all the research work and writing all the text. The other authors have provided valuable feedback and improvement ideas for the paper.

Publication IV: Linking Business and Requirements Engineering: Is Solution Planning a Missing Activity in Software Product Companies?

The author of this thesis is the principal author. She is alone responsible for conducting the research of Phase 1 and writing all the texts related to that. The author is also responsible for summary, literature review, and conclusions for the entire paper. The author of this thesis also conducted the research activities of Phase 2 with Komssi and Kauppinen. The results related to Phase 2 are introduced in Section “Make solution planning visible” (4.5), of which the author of this thesis wrote subsections “Customer segments” (4.5.1) and “Solution” (4.5.3). Kauppinen wrote the other sections, except Section 4.5.5, which was written by Komssi. Vähäniitty provided valuable feedback related to the entire paper.

Publication V: From Feature Development to Customer Value Creation

The author of this thesis conducted the research with other authors and is the main responsible for Sections 2 “Customer value” and practices 2 and 3 in Section 4.3 “Practices that support value creation”. In addition, she provided comments for the entire paper.

1. Introduction

For a software development company, product development is an investment that should provide maximal added value (Boehm 2003; Penny 2002a). Providing value for different customer segments by means of the product is a lifeline for the sales of the product, and via that, to the business of the company. Understanding what customer value means, and how to create value for a large number of customers, however, is not trivial in practice.

From the product development viewpoint this means that a company needs the ability to implement the most valuable requirements in a software product in each product release. Especially in the software product business, the role of the successful selection of the feature enhancements (i.e., requirements) for product releases is recognized as extremely important (Gorchels 2000; Penny 2002a). Market-driven requirements engineering (RE), however, seems to entail special challenges, for example in requirements prioritization and release planning (Karlsson et al. 2002).

This thesis investigates the current state of long-term product planning and requirements prioritization (i.e. feature and requirements selection on different levels), and their linkages to customer-value creation in market-driven software product development. The focus is on software product development organizations. In other words, the thesis concentrates on organizations that develop software-intensive products for a large market of customers and users.

In this section, the thesis's background is covered by describing why long-term product planning and requirements prioritization are important, and by describing the state of the art in the area. Section 1.1 outlines the background and motivation. Section 1.2 defines the research problem and questions. Finally, Section 1.3 introduces the structure of the thesis.

1.1 Background and motivation

According to Boehm (2003), the ultimate sponsors of the project expect that the project's end result will add more value for them than they are paying the project team to create it. On a high level, this means that companies expect their product development organization to add more value to them than they invest in product development.

The purpose of requirements engineering activities is to add business value that is accounted for in terms of software product's return on investment (Barney et al. 2008). The need to make business-based product development decisions means that a company needs the ability to connect business management and software development (Rautiainen et al. 2003). Only by integrating upstream (that is, long-term product planning) and downstream (that is, software development) processes, can value-based decisions concerning the future features of the products be made (Ebert 2005).

Long-term product planning (typically called roadmapping (Kappel 2001; Phaal et al. 2003)) is one approach that companies have used to bridge the gap between business planning and product development. Roadmapping is widely used as a technique in the manufacturing industry (Phaal et al. 2001). The application of the roadmapping approach in the software engineering field is rather new and has not been investigated that much. Additionally, the practical implications of long-term product planning in software product companies in terms of the state of practice or of good practices are not systematically studied.

Prioritizing requirements is also recognized as an important activity to ensure value provision in product development (Karlsson et al. 1997b; Karlsson et al. 2004; Regnell et al. 2001; Siddiqi et al. 1996). The origins of the importance of prioritization are in limited product development resources because time and money are finite in practice. When customer expectations are high and timelines short, the product must deliver the most essential functionality as early as possible (Wiegers 1999). However, the scope of each release must be limited (Siddiqi et al. 1996). The challenge is therefore to select the 'right' requirements out of a given superset of candidate requirements so that all the different key interests, technical constraints, and preferences of the critical stakeholders are fulfilled and the overall business value of the product is maximized (Ruhe et al. 2002).

Requirements prioritization is recognized as a very challenging activity (Carlshamre 2002; Karlsson et al. 2004; Karlsson et al. 2007a). Already, in the early 1990s, Lubars et al. (1993) reported that none of the companies in their study really knew how to assign and modify priorities or how to communicate those priorities effectively to project members. Since then, a good deal of research has taken place in the requirements prioritization area (Ngo-The et al. 2005) and many approaches to requirements prioritization have been introduced in the literature (Karlsson et al. 1997b; Regnell et al. 2001). However, there still is not a silver bullet to requirements prioritization.

Some rationales for the challenges involved in requirements prioritization have been reported in the earlier studies. It is widely accepted that requirements prioritization involves complex decision-making (Karlsson et al. 1997b; Moisiadis 2002). In order to prioritize requirements successfully, domain knowledge and estimation skills are required (Karlsson et al. 2004). In addition, requirements depend on each other and priorities are always relative. An important requirement in one release or to a certain customer may not be as important in the next release or to another customer (Aurum et al. 2003; Carlshamre 2002). Political- and people- related issues are discussed, too (Aurum et al. 2003; Damian 2001; Wiegers 2003).

For companies producing packaged software, the long-term planning and prioritization of requirements are even more challenging than for companies operating in project business. According to Sawyer (2000), the key differences between characteristics of packaged (market-driven) and bespoke software development concern stakeholding and schedule constraints. For requirements engineering this means that in the development of packaged software the future requirements of the software cannot be negotiated with just one or a few customers. Instead, requirements engineering decisions such as the prioritization of potential requirements to be implemented must be made within the company and be linked to the business decisions of the company (Sawyer 2000). In addition, time-to-market is, for many software packages, a survival attribute (Novorita et al. 1996). The normal response to schedule slip in these market-driven cases is to concentrate resources on meeting the most critical requirements with minimal delay (Sawyer 2000).

The idea introduced above means that market-driven companies need effective and business-driven long-term product planning and requirements prioritization practices if they are to survive and to provide value. Unfortunately, in research on RE, the viewpoint has mostly been that of bespoke software development (Karlsson et al. 2002), while RE-related activities outside of projects (e.g., long-term product planning) have had less emphasis. The requirements prioritization research has also focused mostly on the requirements prioritization techniques (Pergher et al. 2013), and not, for example, in understanding the general phenomenon and challenges related to it. Furthermore, the role of long-term planning and requirements prioritization in value creation are not yet much investigated.

1.2 Research goal and questions

The research goal of the study is to investigate both the current state of requirements prioritization and long-term product planning and their linkages to customer-value creation in market-driven software product development.

The detailed research questions of the study are as follows:

1. What is the state of the practice in long-term product planning and requirements prioritization?
2. What are the practical challenges in long-term product planning and requirements prioritization?
3. How do requirements prioritization methods introduced in RE literature suit for solving practical challenges?

Which practices support linking long-term product planning and requirements prioritization to customer value creation?

Table 1 presents the relations between research questions and publications, and Table 2 presents each publication's research objectives. In Table 1 'X' refers to a strong effort on the publication to the related research question, and 'x' refers to a smaller effort.

Table 1. Relations between research questions and publications (X = strong effort, x = smaller effort)

Id	Research question	Publication				
		I	II	III	IV	V
1	What is the state of the practice in long-term product planning and requirements prioritization in practice?	X	x	x	X	
2	Wht are the practical challenges in long-term product planning and requirements prioritization?	X	x	x	X	
3	How do requirements prioritization methods introduced in RE literature suit for solving practical challenges?		X			
4	Which practices support linking long-term product planning and requirements prioritization to customer value creation?				X	X

Table 2. Research objectives of the publications

Id	Name of publication	Objective of the original article
I	Requirements prioritization challenges in practice	Clarify the field of requirements prioritization.
II	Suitability of requirements prioritization methods for market-driven software product development	Investigate how requirements prioritization methods from the RE literature suit for market-driven software product development.
III	Linking business View to requirements engineering: Long-Term product planning by roadmapping	• Provide information about roadmapping in a software-engineering context. • Shed light on challenges that a software-product company introducing the roadmapping approach may face.
IV	Linking business and requirements engineering: is solution planning a missing activity in software product companies?	• Describe the current practices and characteristics of long-term product and solution planning in the software product development context. • Investigate practices that support linking business decisions to RE.
V	From feature development to customer value creation	Analyse the role of RE practices in customer value creation, especially from the perspective of practice.

1.3 The structure of the thesis

The remainder of the thesis is organized as follows. Section 2 provides an overview of prior literature that is relevant to this work. The research methodology is presented in Section 3. Results are summarized in Section 4 and compared with previously published work in Section 5. Thereafter, the contributions of the work are presented and further research areas are presented in Section 6. The five publications annexed to the thesis follow as appendices.

2. Review of the Literature

2.1 Customer value creation

2.1.1 Customer value

Customer value has many meanings in the literature, but two starting-points dominate: value for the customer (customer perceived value or customer received value) and value for the firm (value of the customer, customer lifetime value) (Smith et al. 2007). In this thesis, the basic viewpoint on customer value is the former - customer's perceived value.

Many authors have acknowledged the difficulties involved in actually defining customer value (e.g. (Woodruff 1997)). These difficulties stem from the subjectivity and ambiguity of value, which is compounded by the fact that customer value is a dynamic concept that evolves over time (Naumann 1995). Furthermore, in different disciplines, the value concept is multifaceted and complicated by numerous interpretations, biases, and emphases (Huber et al. 2001; Sharma et al. 2008).

Common for many definitions of customer value is that the concept is related to the trade-off between perceived benefits (what the customer receives) and sacrifices (what he or she gives up) to acquire and use a product according to the customer's perception (Woodruff 1997). In order to truly analyse the customer value of the product, the benefits must be related to sacrifices a customer faces to get and use the product. Perceived benefits can be defined as "a customer's perceived preference for, and evaluation of, those product attributes, attribute performances, and consequences arising from use that facilitates or blocs achieving the customer's goal and purposes in use situations"(Woodruff 1997), not just product features.

The narrowest definitions see "customer value" as the level of return on the product benefits for a customer's payment in a purchase exchange (Normann et al. 1993). Wider definitions are not limited to monetary sacrifices, but assert that the judgment of value results from a trade-off between what the customer receives (e.g., quality, benefits, worth, utilities) and what he or she gives up to acquire and use a product (e.g., price, sacrifices) (Woodruff 1997). According to Smith and Colgate (2007) it is unclear whether customer value is a summative (benefits less sacrifices) or ratio (benefits divided by sacrifices).

In addition to benefits and sacrifices, there are two other typical characteristics in the definitions of 'customer value' (Woodruff 1997). First, customer value is inherent in or linked through the use to some product. Value is not in the products as such, but when the customer makes use of it (Grönroos 2007). This also implies that customers judge products differently in different contexts, and may perceive value differently during purchase and during subsequent use (Gardial et al. 1994). Second, value is something perceived by customers rather than objectively determined by a provider. This means that value has a non-objective nature and that the service provider is in the role of supporting a customer's own value creation (Grönroos 2007).

Ingredients of the value refer to the components that build up the total value. Understanding the customer value of the product requires analyzing these value components. However, it is not trivial to analyze and measure the different components of benefits and sacrifices in practice. Holbrook summarizes customer value as an "interactive, relativistic preference and experience" (Holbrook 2006), which is a bit difficult to understand and apply, but is seemingly intended to capture some of the key characteristics of customer value. These include the following (Ulaga 2003):

- It is perceived uniquely by individual customers;
- It is conditional or contextual (depending on the individual, situation, or product);
- It is relative (in comparison to known or imagined alternatives);
- It is dynamic (changing within individuals over time)

Furthermore, according to Grönroos (2007), (total perceived) customer value has a transaction value component and a relationship value component. This means that customers gain benefits both in short term transactions (e.g. features that I use today) as well as in the long term (e.g., when having a working relationship with a service provider there is no need to search for a new one). Similarly are the sacrifices both short-term (e.g., price) as well as long-term (e.g. how many persons are needed in the cooperation with the service provider). Long-term sacrifices can be divided into three categories (Grönroos 2007):

- Direct relationship costs that depend on the internal systems that the customer has to maintain because of the solution offered by the provider.
- Indirect relationship costs are due to the amount of time and resources that the customer must devote to the maintaining the relationship in case it does not function, as it should.
- Psychological costs are caused when the customer needs to worry about the relationship with the supplier.

2.1.2 Value creation

According to Slater (1997), firms exist to create value for others where it is neither efficient nor effective for buyers to attempt to satisfy their own needs. Many marketing strategists and industrial organization economists emphasize that creation of superior customer value is a key element for companies' success (see e.g. (Porter 2004)). Customers do not look for products or services per se; they look for solutions which they can use so that value is created for them (Grönroos 2007).

Knowing where the value resides from the standpoint of the customer has become critical for suppliers (Ulaga et al. 2001) because greater levels of customer satisfaction lead to greater levels of customer loyalty, positive word-of-mouth, and a stronger competitive position (Bearden et al. 1983; Fornell 1992). Customer value is considered central to both competitive advantage and long-term success of business organizations (Khalifa 2004)

Customer value is perceived and created in customers' value-generating processes, when individual consumers or industrial users make use of the solutions or package they have purchased (Grönroos 2007; Normann et al. 1993). This value-in-use nature of customer value implies that the product value is not developed and embedded into the product or service itself (Vargo et al. 2004). One of the greatest challenges for the industrial markets is to incorporate the "voice of the customer" into the design of new products and services (Haar et al. 2001)

The service or physical product has to fit the customer's value-creation process (Grönroos 2007). Any solution that does not fit customers' process will not interest the firm, because it does not create the value they want. According to Grönroos (2007), customers' everyday activities and their value-creation processes are the most important things for a company to know about its customers.

According to Ulaga and Chacour (2001) value is related to competition. Similarly, MacMillan and McGrath (1997) point out that a company has the opportunity to differentiate itself at every point where it comes in contact with its customers – from the moment customers realize that they need a product or service to the time when they no longer want it and decide to dispose of it. According to MacMillan and McGrath (1997), the company has to take a comprehensive look at the customer's processes. They recommend that companies perform the analysis of the customer lifecycle for each important customer segment.

As early as the 1950s, even before the marketing concept made its way, L.D. Miles at General Electric Corporation developed a set of techniques called value analysis. These techniques aimed at identifying and removing unnecessary costs but still accomplishing the functions that the customer needed and wanted (Miles 1961). It may not be possible to measure how a particular customer assesses the value of a product (the value proposition) at a particular point of time (Smith et al. 2007). However, it is critical for (these) technology-based companies to gain an accurate understanding of the potential value of their offerings and to learn how this value can be further enhanced (Parasuraman 1997; Woodruff 1997; Woodruff et al. 1996).

2.1.3 Value creation in software engineering

Traditionally, much of the research and practice in the software engineering discipline has been done in a value-neutral setting (Boehm 2003). In contrast to a value-neutral approach, Boehm (2003) introduced Value-based software engineering (VBSE), which promotes approaches that maximize the business value for software organizations and provides a systematic process for analysing customer value. Boehm's agenda for VBSE has also initiated further studies for multi-perspective value approaches within requirements engineering (RE). For example, Aurum and Wohlin (Aurum et al. 2007) borrow fundamental aspects of value from economic theory and construct a multilevel model of value alignment from business, product and project perspectives. This value based approach to RE is further applied in a value-based release planning context while studying stakeholders' perspectives and the values influencing requirement selection (Barney et al. 2008).

The "Agile manifesto" (Beck et al. 2001) underlined that the software development process is actually a value creation process. Racheva et al. (2010a) interpret and summarize the manifesto, stating that the main purpose of an agile project is to 1) deliver maximum business value for the client and that 2) agile approaches deliver business value fast and early in the project.

Precisely defining business is not easy. At a high-level, business value can be defined, for example, as something that delivers profit to the organization paying for the software in the form of an increase in revenue, an avoidance of costs, or an improvement in service (Patton 2008). However, there is no single, easy definition of business value in software engineering, as it is rather more slippery and volatile than the most of the authors seem to implicitly assume (Racheva et al. 2009). Khurum et al. (2013) also discuss the lack of a consolidated view on value. They state that there is no complete picture of the value constructs relevant to the different perspectives, required for making software product management and development decisions.

Racheva et al. (2009) were, however, able to identify some characteristics to business value

- Business value in practice tends to be qualitative
- Business value tends to be subjective
- The sources of business value drive requirements prioritization
- Business value of the IT solution requires a degree of trust
- The business value an IT solution tends to be dependent on non-IT business processes

Furthermore, Racheva et al. (2009) found that most studies in their systematic literature review related to business value as understood from the client's perspective, and most often the "customer" here referring to a multi-stakeholder setting in a client organization. These characteristics of value mean, for example, that value is not absolute. Racheva et al. (2010b), for example, observed that there is a link between perceived value and price of implementation. This means that the perceived value of a feature for the customer seems to depend on how much development effort the implementation of it requires. They also found that when making requirements prioritization decisions at inter-iteration time almost all participants stress the importance of what they call a 'negative value' (Racheva et al. 2010b). This term refers to the cost or disadvantage if the requirements are not implemented.

Furthermore, Patton (2008) states that "the business doesn't get its value unless the software is used. In fact, focusing on getting the business its return will compel you to focus on the people who will eventually use the software". This means that defining value for a single feature is difficult during product development, as the value actually is not realized until the end-user uses the product.

The agile frameworks introduced in the literature aim to create value. For example Scrum focuses on frequently prioritizing requirements to maximize Return on Investment (ROI) (Schwaber 2004). According to Barney et al. (Barney et al. 2008), however, the understanding, application and measurement of value from a customer's perspective is still seen as a major problem in the software industry. For example, Racheva et al. (2009) could not find a study in which it is clearly indicated how exactly agile practices create value and keep accumulating it over time. A later study by the same authors also revealed that project participants almost never use an explicit and structured approach to guide value creation throughout a project (Racheva et al. 2010a). Similarly Murtazev et al. (2010) state that "existing value-based approaches do not prescribe how to make whole software processes value based, rather they focus on making specific phase of the process value-based".

The nature of software makes value creation even more complicated (Wohlin et al. 2005). Since software is easily changed and released in several releases, there is a constant trade-off between short-term business goals to satisfy customers and different markets, and the long-term evolution, both of which have to be taken into account to ensure that the software product is competitive in the short- and long-term (Wohlin et al. 2005a).

According to Racheva et al. (2009) a key to value creation in agile projects is decision-making that takes place at the inter-iteration time, when requirements are reprioritized in the face of project uncertainties. According to Bavani (2010), value creation, however, is not a number-crunching approach aimed at measuring cost savings and profit margins alone, but significantly a people-oriented approach. Building a culture of value creation is necessary to uplift practitioners in order to implement successful initiatives (Bavani 2010). Racheva et al. (2009) state as the key implication of their study, that there is a need to pursue a study of value creation in agile projects by deploying empirical research methods and complement it by guidelines for better customer involvement as well as by developing structured methods that will enhance the value-creation in a project.

Racheva et al. (2010b) found that, in practice, some explicit and fundamental assumptions of agile requirement prioritization approaches, as described in the agile literature on best practices, do not hold in all agile project contexts in their study. These are (i) the driving role of the client in the value creation process, (ii) the prevailing position of business value as a main prioritization criterion, (iii) the role of the prioritization process for project goal achievement (Racheva et al. 2010b).

In this thesis, we attempt to understand the value creation in a software engineering context and the role of long-term product planning and requirements prioritization in this value creation. The companies in this case study are not purely agile and the focus of this study is not on agile practices in particular, but on software development in general.

2.2 Software product development

2.2.1 Software product business

Software companies can be roughly divided into two: companies operating in the software project business (other terms used are 'bespoke software' and 'software services') or in the software product business (also called market-driven software development) (Carmel et al. 1998; Hoch et al. 1999; Nambisan 2001). The distinction between a products company and a services (or project) company, however, is not always clear and in many cases hybrid solutions companies (a combination of both models) can be effective in generating a steady stream of revenues and profits (Cusumano 2004).

According to Nambisan (2001), software companies moving towards the software product business from custom-made solutions face new managerial product development challenges. According to Sawyer (2000), two major differences between bespoke software development and market-driven software development concern the characteristics of stakeholding and schedule constraints. In market-driven software development, there is significant pressure on time-to-market and the software product is often offered to a market through recurrent releases.

2.2.2 Requirements engineering

Requirement and feature

Wiegers (2003) defines a *requirement* as a property that a product must have to provide value to a stakeholder. Requirements engineering covers all of the activities involved in discovering, documenting, and maintaining a set of requirements for a computer-based system (Kotonya et al. 1998). "Requirement", however, is not an unambiguous term, as different authors seem to define it differently and emphasize different viewpoints in their definitions. For example, an IEEE standard (IEEE Std 610.12 1990) defines the term as,

- a condition or capability needed by a user to solve a problem or achieve an objective.
- a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document; or
- a documented representation of a condition or capability as in (1) and (2).

Davis (1993) supplements the IEEE's definition by defining a requirement as "a user need or a necessary feature, function, or attribute of a system that can be sensed from a position external to that system". Kotonya and Sommerville (Kotonya et al. 1998) state that (system) requirements define what the system is required to do and the circumstances under which it is required to operate.

Feature represents a logical unit of behaviour from the perspective of one or several stakeholders in the product and is generally used to group requirements (Bosch 2000). In this thesis, the term is used to distinguish (usually not-yet-defined-in-detail) higher-level requirements from more mature and defined requirements.

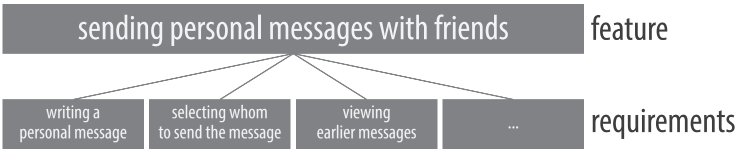

sending personal messages with friends
feature
writing a personal message
selecting whom to send the message
viewing earlier messages
...
requirements

Figure 1. A feature represents a logical unit of behaviour from the stakeholder's perspective

Figure 1 illustrates via an example the distinction between the concepts 'feature' and 'requirement'. In this thesis, the term 'requirement' is used when the size or maturity of the requirement does not play a role. The term 'feature' is used only in those cases where the authors want to emphasize the large and undefined nature of the requirements they are addressing.

Requirements engineering process

The requirements engineering process can be defined as a structured set of activities, which are followed to derive, validate, and maintain a systems requirements document (Kotonya et al. 1998). According to Kotonya and Sommerville (1998), the basic sequence of the requirements engineering process includes requirements elicitation, requirements analysis and negotiation, requirements documentation, and requirements validation. However, there is no single requirements engineering process which is right for all organizations. In addition, the sequential process description per se is not enough (Kotonya et al. 1998). Sound requirements processes emphasize a collaborative approach to product development, involving multiple stakeholder perspectives in a partnership through a project (Wiegers 2003).

Requirements engineering is usually seen as the first phase of the development cycle. For example, Jackson (1995) argues that requirements engineering and design are separate activities, because requirements are mostly concerned with the problem to be solved and design is concerned with the solution to the problem. However, Kotonya and Sommerville (1998), for example, argue that these two are interrelated activities.

Requirements engineering should be seen neither as a single activity at the beginning of a project nor a purely sequential process. Instead, requirements engineering is needed throughout the product development life-cycle and iterations are usually needed (Wiegers 2003). An interesting point to mention here is that Royce's (1970) waterfall article actually explored how the initial waterfall model could be developed into an iterative model, with feedback from each phase influencing previous phases. Ironically, only the initial model received notice and his criticism of this initial strictly sequential model has been largely ignored.

2.2.3 Market-driven requirements engineering

Market-driven requirements engineering refers to requirements engineering in companies operating in the software product business. Usually, the viewpoint in the requirements engineering literature has been that of bespoke software development (Karlsson et al. 2002). As a whole, requirements engineering outside projects seems not to have been discussed that much in the literature. According to Ebert (2005), this might be because of its complexity (e.g., overlapping ownerships) and the historical division between product management and requirements engineering, which was perceived as an internal engineering discipline.

However, the differences in the business models are reflected in requirements engineering as well. For example, Karlsson et al. (2002) found that requirements engineering for commercial off-the-shelf software packages entails special challenges. In companies operating in the product business, requirements engineering is needed not just within projects, but also before projects (Ebert 2005). This means that in a market-driven situation the traditional, monolithic requirements specification is of limited value when managing a steady stream of incoming requirements of varying quality (Karlsson et al. 2002).

One of the key effects that the movement to a product business has had on requirements engineering is the increasing importance of long-term product planning and requirements prioritization. In market-driven software product development, wide markets with a large customer base outside the company and lots of stakeholders within the company are involved. This means that the future development steps of a product cannot be negotiated with just one or a few customers. These decisions must be made more strategically and within the company, and with the developer bearing all the financial risks involved (Sawyer 2000). This brings up also issues like a need for common language between business decision-makers and IT decision-makers (Paech et al. 2008). In order to provide value with their products, companies need to place an emphasis on the selection and prioritization of requirements before projects in addition to within-project activities (Ebert 2005).

For example, Sawyer points out that when the software is offered through recurrent releases, careful release planning and requirements prioritization are needed (Sawyer 2000). According to Moisiadis (2002) 'the notion of releasing progressive versions and updates of products, as well as the rising demand on developers to build systems that go to market much quicker than ever before, has led to the need to prioritise requirements at the earliest possible stage in the systems development life cycle'. Wiegers (2003) suggests that priorities should be evaluated and adjusted periodically throughout development as customer needs, market conditions, and business goals evolve.

According to Ebert (2005), only by integrating upstream (e.g. roadmap) and downstream (i.e. project) processes will projects be successful. Requirements engineering decisions (e.g., prioritization) must be linked to business decisions of the company (Rautiainen et al. 2003). For example, Favaro (2002) points out that because the purpose of the requirements process is to add business value, the person in the position of managing requirements is also in the position of making the most of strategic opportunities. Thus, a central activity within the software vendor organization is deciding when the next releases of their software products should be made generally available and what feature enhancements (i.e. requirements) they should contain so as to maximize future revenue (Penny 2002b).

Karlsson et al. (2002), however, reported that many of the key challenges in market-driven requirements engineering had a link to the selection and prioritization of requirements. Such challenges were making trade-offs between the requirements demanded and new, inventive requirements, the gap between marketing staff and developers, bad time estimates having an effect on release plans, and requirements overload complicating release planning (Karlsson et al. 2002). Also Ebert (2006) points out that a major obstacle of a product manager is the need to balance a variety of needs from markets, customers or stakeholders and aligning them into an optimized allocation of spare resources. Furthermore, he argues that little specific guidance is available in the literature for software product management (Ebert 2009) which is supported also by case studies showing that creation of software product value through requirements prioritization decision-making is only partly understood (Petersen et al. 2009).

2.3 Product planning

Wiegers (2003) defines a requirement as a property that a product must have to provide value to a stakeholder. In order to maximize the value provided by a product, a successful selection of the requirements to be implemented is needed. When customer expectations are high, timelines short, and resources limited, the most essential functionality of the product should be delivered as early as possible (Wiegers 1999). However, the scope of each product release must be limited (Siddiqi et al. 1996). The challenge (in product development) is to select the 'right' requirements out of a given superset of candidate requirements so that all the different key interests, technical constraints, and preferences of the critical stakeholders are fulfilled and the overall business value of the product is maximized (Ruhe et al. 2002).

As stated earlier, the goal in the software product business is to sell the packaged software without customer-specific modifications. This means that the development organization is alone responsible for deciding which requirements to implement and selecting the stakeholder representatives. Thus, the developer bears all the financial risks included and there is no single customer who is the principal stakeholder, as with bespoke software development (Sawyer 2000).

In this thesis the focus is on three product planning activities, roadmapping, release planning and requirements prioritization. Even though these activities are presented separately in this thesis, is worth mentioning that in practice these activities are typically overlapping and cannot be totally distincted from each others.

2.3.1 Long-term product planning (Roadmapping)

Requirements documents for short projects are not sufficient to ensure the comprehensive understanding in the organization that developing software products for wide markets requires (Karlsson et al. 2002). Long-term product planning is one route that software product companies follow to explicate the link that is needed between business decisions and requirements engineering. Weak, or unclear, visions or goals for the software development (due to not being communicated or not being clear enough) contributes to weak communication, primarily, between those defining the requirements and the development unit because there is no mutual understanding of the goal (Bjarnason et al. 2010).

Roadmapping can be defined as a flexible technique that is used to support strategic and long-range planning (Kappel 2001). The basic purpose of roadmapping is to explore and communicate the dynamic linkages between markets, products, and technologies over time (Kappel 2001). A typical roadmapping process gathers together stakeholders from different functions of the organization to plan and make decisions, and provides the roadmap as an outcome for presenting the decisions that have been made.

The output of the roadmapping process is called a roadmap (see Figure 2). The generic roadmap is a time-based chart, comprising a number of layers that typically include both commercial and technological perspectives (Phaal et al. 2004).

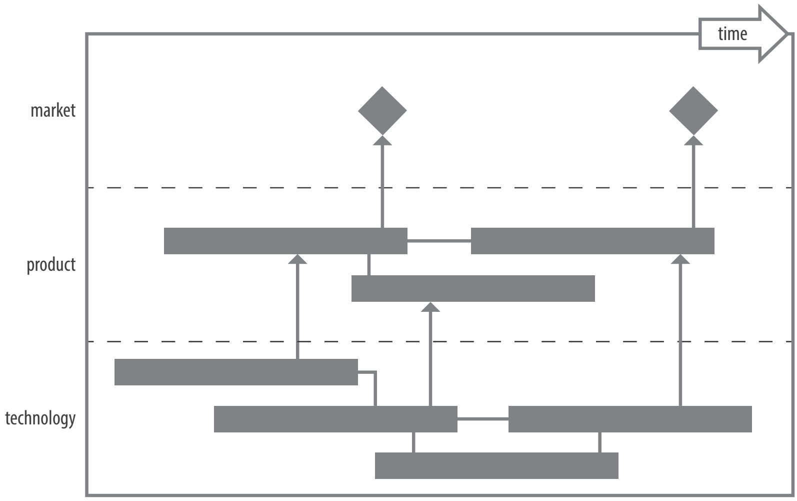


Figure 2. A generic roadmap is a time-based chart, comprising number of layers (Phaal et al. 2004)

Roadmaps communicate visions, attract resources from business and government, stimulate investigations, and monitor progress (Galvin 2004). According to De Gregorio (2000), a roadmap should provide a simple but powerful visualization of a forecast. Roadmaps, however, have a dual nature; they are both forecasts and plans (Kappel 2001). By forecasting he means that roadmaps articulate what is likely to happen and by plans that roadmaps usually also articulate the course of action.

Roadmapping is reported as a tool for widening the focus from short-term plans to more long-term thinking. According to Albright and Kappel (2003), today's business climate can lead to a focus on short-term thinking, often tied to the reporting needs of the budget cycle or the next deliverable. Roadmapping helps to focus on long-term planning and on the highest-priority topics (Albright et al. 2003). According to Phaal et al. (2003), roadmaps provide a means of charting a migration path between the current state of the business (for each layer) and the long-term vision.

Roadmaps can be used for different purposes (De Gregorio 2000). This means that roadmaps articulate different aspects of strategy and are snapshots from different viewpoints. In addition, the information described in the roadmaps can vary according to the abstraction level and type of information (Phaal et al. 2003). This means that roadmaps can contain information from early forecasts to documented business decisions. For example Komssi et al. (2011) propose that an analysis of customer value and customers' processes should be integrated into roadmapping. In addition, sometimes just a high-level view of the topic is needed and in some cases detailed information is in place. Roadmaps also vary greatly according to their presentation formats. Sometimes just one picture is needed, but there are cases where documentation of over one-hundred pages may be reasonable (Phaal et al. 2003).

Many of the benefits of roadmapping, according to the literature, are derived from the roadmapping process rather than the roadmap itself (Kappel 2001). The process brings people together, allowing them to share information and perspectives. According to Phaal et al. (2003), roadmapping also provides them with a vehicle for the holistic consideration of problems, opportunities, and new ideas. Albright and Kappel (2003) point out that the use of cross-functional teams improves communication and the ownership of plans by providing a common vocabulary and combining each member's special knowledge. However, the key challenges identified in roadmapping are how to establish a suitable roadmapping process for the organization and how to keep roadmaps alive and updated (Vatananan et al. 2010).

The graphical form of the roadmap is reported as being a powerful communication mechanism (Phaal et al. 2003). Compared to merely textual documents, graphical roadmaps usually give a better high-level view and help in sharing the common understanding with different stakeholders. Vähäniitty et al. (2009) suggest that product roadmap visualization should express the release and development schedules for the product, composition of individual releases, changes to the underlying technology, services requiring attention from product development and planned resource usage, while project management tracks how successfully the roadmap is being acted on.

In the case organizations of this study, the term roadmapping is most often used to describe the activities that aim to allocate the potential future features of the products in forthcoming releases. Thus, roadmapping can be seen as a product management level prioritization activity because the aim of it is to prioritize features to releases over time.

2.3.2 Release planning

Requirements engineering for software products can be generally seen as a way of synchronizing the work with the continuous flow of candidate requirements and the work with the discrete release events (Regnell et al. 2005). Requirements are received and registered on a continuous basis by the product manager from all kinds of submitters internal or external to the company, such as customers, sales representatives, or development teams (Regnell et al. 2005). The gathered requirements are then typically stored into a *requirements repository* that is dynamically evolving with data of varying types and levels of abstraction (Regnell et al. 2005).

Release planning is the activity through which the requirements in the repository are realized into a product releases. Hence, the input for the release planning process is a set of requirements that are evolving over time due to changing user needs and better problem understanding (Ruhe 2010). During release planning the product manager communicates with other roles in the development team: project manager, software engineers, testers, technical authors, translators, marketing, etc. (Regnell et al. 2005).

The challenge in release planning is to select the 'right' requirements out of a given superset of candidate requirements so that all the different key interests, technical constraints and preferences of the critical stakeholders are fulfilled and the overall business value of the product is maximized (Ruhe et al. 2002). Furthermore, over-scoping of releases is reported as a common problem because of issues such as continuous requirements inflow, lack of consolidated view of capacity, lack of development team involvement in early phases, requirements not being agreed with the development team, unclear vision or goal and detailed requirements specifications developed upfront (Bjarnason et al. 2010).

Different release planning methods have been proposed during the last decade (see (Ruhe 2010) for overview) including:

- *Greedy release planning* (Cormen et al. 2006) that uses an algorithm designed to make a locally optimal choice at each stage in selecting requirements based on their priority and the resource consumption.
- *Optimizing value and cost* (Jung 1998), that applies an algorithm to balance the cost and value of the requirements, and then implement the most cost-effective set.
- *Combining optimized value and cost with the interdependencies of requirements (Carlshamre 2002)*, that selects requirements based on the trade-off between a requirement's value and cost while considering the interdependencies between requirements.
- *The next release problem (Bagnall et al. 2001)* that looks exclusively at the cost per feature. The objective of planning is to ensure that the demands of a company's client base are satisfied as much as possible while at the same time ensuring that they themselves have the resources to undertake the necessary development. There is no involvement of stakeholders in the prioritization of requirements.
- *The incremental funding method* (Denne et al. 2004) that aims at delivering functionality in chunks of customer-value features, sequenced to optimize the project's net present value. The method is focused on the maximization of the overall financial value.
- *EVOLVE (Greer et al. 2004)* that is an evolutionary and iterative approach for software release planning that looks ahead for more than one release. The method tries to balance the conflicting stakeholder opinions to achieve the highest degree of satisfaction with the resources available. The method generates more than one alternative, each of which represents a trade-off between fulfilment of stakeholder expectations and the total benefit of the proposed plans.
- *Software product release planning through optimization and what-if analysis* (Van den Akker et al. 2008) that attempts to apply mathematical programming to provide a solution for the next release problem.
- *CSP modelling (Regnell et al. 2011)* where relative and absolute priorities, interdependencies, and other constraints are expressed

as relations among variables representing entities such as feature priorities, stakeholder preferences, and resource constraints.

- *CERP (Heikkilä et al. 2010a)* that facilitates the active involvement of stakeholders in the different stages of the planning process
- *The multi-criteria method (Fernandes et al. 2008)* that evaluates the value of XP release plans, which is based upon software minimum marketable features, information economics, risk analysis, and stochastic modelling

2.4 Requirements prioritization

2.4.1 Definition of requirements prioritization

Sommerville (2010) defines requirements prioritization as an activity during which the most important requirements for the system are discovered. Requirements prioritization is needed not only to ignore the least important requirements, but also to improve effective project management. Clearly defined requirement priorities are the essential basis for conflict solution among requirements, for decision-making during architectural design and for test case prioritization (Herrmann et al. 2006). Thus, Harwell et al. (1993) describe a priority as being a characteristic of a requirement that can be used for different purposes, depending on program and company needs. Prioritization can be viewed as a sub-problem of release planning, where the latter not only involves assigning priorities according to a set of criteria reflecting the views of a set of stakeholders, but also includes scheduling, resource planning and taking into account requirements interdependencies (Carlshamre et al. 2001).

According to Boehm (2003), 'much of current software engineering practice and research is done in a value neutral setting, in which every requirement is treated as equally important'. According to Wiegers (2003), requirements prioritization is needed to indicate how essential each requirement, feature, or use case is to a particular product release. If all the requirements are considered to be equally important, it is hard for the project manager to respond to budget cuts, schedule overruns, personnel losses, or new requirements added during development.

The traditional view of requirements prioritization is to see it as an 'activity in which priorities are given for individual requirements within a software development project'. However, understanding the value of requirements is not just within-the-project activity in software product development (Ebert 2005). Instead, the prioritization of requirements before projects and allocating them to releases is an essential activity in order to provide value for customers and users. Additionally, Wiegers (2003) suggests that priorities are evaluated and adjusted periodically throughout the period of development as customer needs, market conditions, and business goals evolve.

A definition for high-priority requirements in this thesis is as follows: '*Those requirements that provide the largest fraction of the total product value at the smallest fraction of the total cost.'* (based on definitions in e.g. (Jung 1998) and (Karlsson et al. 1997b)).

2.4.2 Challenges in requirements prioritization

Requirements prioritization has been recognized as a challenging activity in the literature. Already Yeh (1992) reported that that requirements prioritization is one of the most crucial and at the same time difficult tasks that faces the decision makers. However, there is little direct research on why requirements prioritization actually is difficult. Still, many authors have provided some rationales for the challenges involved in prioritization. These include the following:

- Taking different aspects into account in prioritization
- Lack of information about aspects
- Linking priorities to business goals
- Changes in the environment, requirements and priorities
- Different opinions about priorities
- Dependencies between requirements
- Prioritizing large amounts of requirements

Taking different aspects into account in prioritization

Requirements prioritization is a decision-making activity where several aspects have an effect on priorities (Benestad et al. 2011). However, there are many challenges related to taking these aspects into account in practice. According to Berander (2007), it is seldom clear which aspects to use in prioritization. Carlshamre (2002) supports this view by pointing out that decision-makers find that it is not easy to define the aspects on the basis of which prioritization decisions should be made. For example, Bubenko (1995) points out that practitioners lack knowledge of how to quantify the benefits and risks of (different alternative designs and) requirements. Also, according to Svensson et al. (2011) it is most common to have no specific or explicit criterion defined when prioritizing (quality requirements).

It is also not easy to combine aspects together as practitioners do not know how to emphasize different aspects compared to the other aspects. Karlsson et al (2002) report problems in market-driven companies in having a good balance between market-driven and technology-driven requirements in their releases. Even if a developer is provided priorities, there are too many complex factors that a developer does not or can not take into account for in order to mitigate the critical risk of not implementing an acceptable subset of the requirements (Park et al. 1999). Benestad et al. (2011) also point out that requirements prioritization models in the literature focus on only select parts of a possibly larger space of relevant planning factors.

One big challenge in taking different viewpoints into account is that the aspects are interrelated. According to Wohlin and Aurum (2005b) it is almost impossible to choose truly independent aspects. Also Ruhe et al point out that when multiple aspects are included in prioritization, the aspects interact so that the changes in one impact another (Ruhe et al. 2003).

Lack of information about aspects

One key problem in prioritization is that decision-makers do not have enough information about the issues affecting their decisions (Boehm et al. 2001; Damian et al. 2003) For example, Boehm et al. (2001) point out that that requirements often must be negotiated among success-critical stakeholders who are often unsure of their own needs and even less so of the needs of others.

One problem is that decision forums often do not include a representative sample of stakeholders (Damian et al. 2003). However, only the stakeholders can properly prioritize the requirements, while only the developers can properly estimate the cost and schedule consequences of the stakeholders' priorities (Wiegers 1999). Sivzattian and Nusebeih (2001) also point out that prioritization approaches introduced in the literature fail to explicitly account for uncertainty and incomplete knowledge of the real world.

Linking priorities to business goals

Incorporating the expertise from marketing departments and the visions of top-level management in the prioritization process is important. For example participants in a study conducted by Barney et al. (2009), business-perspective criteria was the most important for selecting project requirements both currently and in the ideal case. However, one of the main challenges identified by Regnell et al. (1998) is to relate the continuous prioritization of incoming requirements to a long-term product strategy for a range of market segments.

Aligning the links between business goals and requirements has been found as challenging in practice (Khurum et al. 2012). Damian and Zowghi (2003) found out that requirements expressed by customers were often not aligned with business requirements. Also Bubenko (1995) reports that links between business and enterprise models and information system specifications are usually not maintained. Firesmith (2004) points out that it is often difficult to directly relate requirement priority to business goal importance. Interestingly, related to this challenge, Khurum et al. (2012) report that even though e.g. business value of requirements, competitors or target customers are reported as aspects to be considered during prioritization, difficulty in aligning business goals with decisions is reported as the one of the biggest challenges in requirements prioritization.

Changes in the environment, requirements, and priorities

Market-driven companies have to deal with frequent changes in importance of different requirements when competitors improve, market changes, or when customers just change their mind (Karlsson et al. 2002). Stakeholders may for example change their minds once they understand the cost and schedule implications (Wiegers 1999). Even stakeholders (in the process) may change (Firesmith 2004). Not to mention that also needs and individual requirements may change, and they are often incompatible (Firesmith 2004). All this changes also priorities, which makes prioritization difficult.

Different opinions about priorities

Aspects such as importance are unambiguous concepts and dependent on which perspective the stakeholder has (Berander 2007). An important requirement in one release or to a certain customer may not be as important in the next release or to another customer (Carlshamre 2002). Implementation of one requirement bringing value to one stakeholder might cause a negative value proposition to another because stakeholders may have goals that conflict with one another (Nuseibeh et al. 2000). Also Herrmann et al. (2006) mention the differences in utility perception of different stakeholders.

Aurum et al. (2003) describe the RE process, in essence, a complex communication and negotiation process involving many stakeholders. They argue that it includes a great deal of invisible decision-making (Aurum et al. 2003). For example, Wiegers (2003) argues that customers might not want to prioritize their requirements because they are afraid of having just the most important ones done and developers do not want to admit that they are not able to implement all the requirements. Political issues are discussed by other authors, e.g. Andriole (1998), too. Findings of Racheva et al. (2010b) indicate that in practice the developer viewpoint plays a much more important role than what is recommended in the literature.

Dependencies between requirements

Priorities are relative and the requirements may depend on each other in many levels (Carlshamre et al. 2001). Because the value of a requirement may depend on another requirement (Karlsson et al. 1997a), the arrival of new requirements may require the reprioritization of the entire backlog. This means that as new requirements are added, the relative priorities of existing requirements may need to change accordingly (Fellows et al. 1998).

Furthermore, the system may be incrementally developed so that some of the requirements are implemented before others. Thus, the important priorities become the priorities of those remaining requirements that have yet to be implemented (Firesmith 2004). According to Benestad et al. (2011) developing a shared understanding of features and their possible interdependencies requires the most effort in the planning process, and the allocation of features to releases actually constitutes a less complex task. Khurum et al. (2012) found that ‘requirements dependencies’ was one of the two most applicable challenges in the industrial context.

Prioritizing large amounts of requirements

Prioritizing large amounts of requirements has been reported challenging in many publications where requirements prioritization methods are studied (e.g. (Karlsson et al. 2002)). Karlsson et al. (2002) found out that thousands of requirements resulted in difficulties when prioritizing requirements for the next release. Furthermore, Firesmith (2004) points out that techniques for determining the priorities of all requirements, such as pairwise comparisons or Quality Function Deployment (QFD) typically do not scale unless requirements are previously grouped in some manner. The difficulty of prioritizing large amounts of requirements is supported also by Babar (Babar et al. 2011) who points out that there is no evidence of a successful prioritization technique that would solve the problem of a large set of requirements.

2.4.3 Requirements prioritization methods

Requirements prioritization started to gain interest in the requirements engineering research in the nineties, when general RE studies noted the challenges and importance of prioritization (Lubars et al. 1993). In the late nineties, authors also started to introduce methods for prioritizing requirements (e.g.(Beck 1999; Karlsson et al. 1997b; Wiegers 1999)), which continued in to the twenty-first century (e.g. (Berander et al. 2006a; Herrmann et al. 2008; Lauesen 2002; Leffingwell et al. 2003).

The literature offers several methods for requirements prioritization. As requirements prioritization could be seen as a basic sorting problem of items, in theory any algorithms could be used to put a set of requirements in order. Comprehensive lists of methods and sorting algorithms proposed for requirements prioritization in the literature are presented e.g. in (Herrmann et al. 2008), (Kukreja et al. 2012) and (Racheva et al. 2010b).) In this thesis, we concentrate only on those prioritization methods that are discussed in the requirements engineering literature in the context of prioritizing requirements.

The prioritization methods introduced in the requirements engineering literature vary from high-level prioritization process descriptions to detailed prioritization algorithms (Berander et al. 2006b) and can be categorized in different ways. Such different approaches work on different measurement scales, focus on different aspects, and have different levels of sophistication (Berander et al. 2005). Approaches on different levels typically focus on solving some parts of the requirements prioritization problem and put less emphasis on other challenges (Berander et al. 2006b). One way to categorize methods is for example based on scales that are used in the prioritization.

Different requirements prioritization methods introduced in the literature seem actually to be intended for slightly different purposes. These purposes can be e.g.:

- Sharing limited product development resources and solving conflicts between different stakeholders (e.g., voting, million dollar test)
- Collecting opinions from different user and customer groups about their preferences (e.g. top ten requirements)
- Analysing requirements from different viewpoints (e.g. Wiegers' method, Cost-value approach)

- Recording and communicating further which requirements are important (e.g. IEEE recommendation of priority groups)

Requirements prioritization methods introduced in the literature can be categorized in many ways. In this thesis, prioritization methods are introduced according to the background idea that the methods have for requirements prioritization. These categories are briefly introduced here and examples of methods in each category are given and described. In addition, the benefits and disadvantages of the different types of methods are briefly discussed.

The introduced categories of different requirements prioritization methods are as follows:

- Grouping and sorting methods
- Methods that combine different aspects affecting priorities
- Voting and investing methods

Grouping and sorting methods

In grouping and sorting methods ‹requirements are put in groups or sorted according to their importance or urgency. The exact names of the groups, as well as the amount of them, vary in different approaches, but the basic idea remains the same. Examples of these methods are introduced in Table 2. The best-known implementation of this method is to put requirements in three groups: “Must”, “Essential”, and “Conditional”.

Some authors suggest that practitioners use another approach (e.g., additional calculations) instead of estimation to divide the requirements into priority groups. For example, Dver (2003) suggests using a balanced scorecard approach to categorize product requirements into three rankings: high, medium, and low.

Table 3. Grouping and selection methods

Method	**Brief description**	**Reference**
Priority Groups	The most common implementation of this method is to put requirements into three groups (must, essential and conditional) according to the requirement’s importance to customers.	IEEE (1998)
Grouping with internal grouping	Requirements are first put into groups and then regrouped within the groups as long as the end result is a descending ordered priority list of requirements. Groups can be internally ranked by using other techniques as well.	(Karlsson et al. 1998)
Spanning tree matrix, Bubble sort, Binary Search Tree	Prioritization is performed on the basis of a general sorting algorithm.	(Karlsson et al. 1998)

The grouping techniques are easy to use and learn, as no complex calculation is needed. However, before using these techniques the stakeholders should discuss and explicitly define what the different groups mean in their case.

Methods that combine aspects affecting priority

A set of prioritization methods is based on the idea that the priority of a requirement is a combination of the estimates of values given to different aspects that affect the priority of a requirement. Examples of these methods are introduced in Table 4. The most typical aspects of these methods to be included in the prioritization are the requirement's value for the customer and the implementation costs of the requirement (e.g. (Karlsson et al. 1997b)).

Table 4. Methods that combine aspects affecting priorities

Method	Short description	Reference
AHP (Analytical Hierarchy Process)	All unique pairs of items are compared to determine which of the two is of higher priority, and to what extent.	(Saaty 1980)
Hierarchy AHP	A modification of AHP in which only requirements on the same level of a hierarchy are compared with each other.	(Saaty 1980)
Cost-value approach	AHP-based method in which all possible requirement pairs are compared according to their importance and implementation costs. The percentage share that a requirement has for total value and the total costs of all requirements are calculated for each requirement. (Cost-value approach is one instance of Hierarchy AHP)	(Karlsson et al. 1997b)
Ordinal cost-value approach	Requirements are put into three groups according to their value to customers and into three groups according to their implementation costs. The results are presented in a cost-value scattered diagram.	(Karlsson et al. 2005)
Wiegers' method	Each requirement is evaluated on a scale from 1 to 9 according to its value to the customer, the penalty if it is not implemented, implementation costs, and risks. Priority is calculated by dividing value + penalty by cost + risks.	(Wiegers 1999)
Impact validation	The impact that each proposed requirement has on the achievement of the high-level goals of the project is evaluated on a defined scale. For each requirement an impact sum is calculated. The requirement having the greatest impact is seen as the most important and so on.	(Gilb 2005)
MDRPM (Market Driven Requirement Prioritization Model)	AHP with a consistency check added to the normal procedure.	(Iqbal et al. 2010)
Simulation-based Fuzzy Multi-attribute Decision Making	Model takes the imprecise nature of requirements into account by modelling their attributes as fuzzy variables.	(Ejnioui et al. 2012)

Compared to just grouping the requirements, these kinds of methods provide wider insight into prioritization and may help to take different aspects affecting priority into account better (e.g. (Karlsson et al. 1997b). However, many of these methods require complex calculations and need a lot of effort to perform the prioritization. In addition, even though these methods usually provide much more definite results than grouping (e.g., an individual importance percentage for each requirement), it should be remembered that no result can be better than the estimates given.

Voting and investing methods

The investing methods are based on the idea that there is a certain amount of resources to be invested in the product development in one release and that prioritization is actually allocating these resources between different potential requirements (see examples from Table 5). Voting comes to the question when there are many stakeholders that have different viewpoints on how to make this allocation (e.g., in those cases where several product managers are responsible for different customer segments).

Table 5. Voting and investing methods

Method	Short description	Reference
$100 test / Cumulative voting	Each stakeholder gets an imaginary $100 which she can allocate to requirements as she wants. The requirements that get the most money allocated are the top priority requirements. The amount of 'dollars' can be according to he situation divided evenly or unevenly among stakeholders. For example business units could get 'dollars' according to their investment levels.	(Leffingwell et al. 2000)
Hierarchical Cumulative Voting	Prioritization is performed as in cumulative voting, but not all requirements are prioritized at the same time. Prioritizations are performed at different levels of a hierarchy, and within different groups of requirements in that hierarchy.	(Berander et al. 2006a)
Planning Game	The XP version of Cost-Value approach where the project group estimates for every user story how many programming weeks it will take to implement it and the user decides which user stories she wants implemented first.	(Beck 1999)
Distributed Prioritization Process	Each stakeholder prioritizes candidate list of priorities. Product strategy team forms candidate priorities based on individual prioritizations. Iteration if needed.	(Regnell et al. 2001)
Top Ten Requirements	Each stakeholder selects the 10 most important requirements from her viewpoint. The requirements that are selected by many stakeholders in their top 10 lists are considered the most important.	(Lauesen 2002)

The benefit of using voting and investing is that they serves as a more controlled way to take the opinions of different stakeholders into account. However, the disadvantages of these methods are that they still leave some space for 'politics'. For example, stakeholders may not give their votes to important requirements that are common for every stakeholder bacause they just want to invest in their specialties.

2.4.4 Empirical evidence on requirements prioritization

Several systematic literature reviews have been published in the area of requirements prioritization in recent years (Achimugu et al. 2014; Herrmann et al. 2008; Pergher et al. 2013; Pitangueira et al. 2013). These studies present evidence about requirements prioritization, the nature of studies published in this area, and the challenges that have been reported with existing techniques.

Requirements prioritization has been significantly discussed in the requirements engineering domain (Achimugu et al. 2014). According to Pergher et al. (2013) there is relatively recent interest in the topic "Requirement Prioritization". Most of the papers in that 2013 study had been written in the last 7 years and 37% (little less than the half of them) were not older than 3 years (Pergher et al. 2013).

According to the literature reviews, requirements prioritization is used for several different purposes. Achimugu et al. (2014) summarize based on their study that there are four main purposes of requirements prioritization in practice. These are:

- Determining the relative necessity of elicited requirements
- Negotiation of precise requirements
- Determining the implementation schedule
- Determining the judicious utilization of fund

The research in the area of requirement prioritization seems to be focused on prioritization techniques (Pergher et al. 2013). According to Pergher et al. (2013), the majority of the studies are about the validation of research or solution proposals. Research efforts have been put into proposing and validating new approaches for requirements prioritization (or refinements of previous ones). Less effort has been put into the evaluation, or reviewing process (of the methods) (Pergher et al. 2013).

According to Achimugu et al. (2014), the applicability of techniques in complex and real settings has not been reported yet. Pergher et al. (2013) report that even though there are some empirical studies, they are very often based on case studies within industry, reporting about the application of a specific technique/method. The experiments (comparing many techniques/methods) are mostly performed as experiments with students as subjects (Pergher et al. 2013). Pitangueira et al. (2013) also point out that most studies use experimental data only on a small scale, which restricts the application of results in real situations, on larger scales and generalization.

In empirical studies, the approach seems to usually be bottom-up, centring on the techniques (Pergher et al. 2013). Most of the papers investigate the accuracy of the prioritization technique: this is the main dependent variable of the study aiming at identifying the correspondence of the ranking performed with the perception of the participants. Two other relevant dependent variables are the time required for the prioritization process, and the easiness of use of the approach (Pergher et al. 2013). The functional requirements seem to be the main research focus (Pergher et al. 2013; Pitangueira et al. 2013).

The studies' settings also seem to differ from real industrial situations in many ways. According to Pergher et al. (2013), most of the requirements submitted to subjects are at a very high-level of granularity, being mainly composed by a title and a textual description. We also found in our own literature review that most of the studies have been conducted with "toy" requirements that have been invented for the purpose of the study. Only some cases were performed with real product development requirements. In one study, the prioritized items were not actually requirements. Furthermore, only half of the studies report an example of the requirements used, and in general very few studies discuss the level of granularity and the impact that it can have on the final result.

In the review conducted by Pitangueira et al. (2013), none of the analysed studies distinguished between non-functional and functional requirements, and only a few considered interdependences between requirements. Furthermore, the role of secondary requirements was not investigated. How the quantified contribution of secondary requirements to primary requirements can be used for quantifying benefit of secondary requirements remains an unanswered question (Herrmann et al. 2008). Furthermore, the analysed requirements prioritization studies have been conducted with quite a small number of requirements relative to real product development situations. We were able to find two studies (Carlshamre 2002; Regnell et al. 2001) with 50-1000 requirements in the example. The other studies used 6-25 requirements.

Current requirements prioritization methods seem to suffer from many limitations. Achimugu et al. (2014) report that current limitations with the methods are related to scalability, computational complexity, rank updates, communication among stakeholders, requirements dependencies, error proneness and lack of fully implemented requirements systems. Herrmann et al. (2008) also point out that requirements dependencies are largely neglected in requirements prioritization, despite the fact that the RE community agrees on their importance. Furthermore, Pitangueira et al. (2013) mention that consideration of the interdependence between requirements already being used, however, still requires further elaboration.

It also seems that current requirements prioritization methods take for granted that there are objective values to provide inputs into the methods. Case study findings by Racheva et al. (2010b) give rise to the suspicion that these objective values may not always exist and that they are sometimes very difficult to determine. With respect to value-based decision-making, they observed that the consideration of value as a prioritization criterion is complex. The existence of objective values to feed as input into the prioritization methods is questionable; instead, the priority seems to be a combination of subjective value-based criteria. According to Pitangueira et al. (2013) multi-goal modelling is a growing trend in the current studies; however, there is much room to create models that hew even closer to software engineers' reality. The inclusion of user judgments and new restrictions, such as risks and uncertainties, constitute an open field for exploration.

One interesting finding based on the literature reviews is that it seems to be not clear what the value of using requirements prioritization methods actually is. Herrmann et al. (2008) investigated by a literature review how existing methods approach the problem of requirements prioritization based on benefit and cost. Interestingly, their analysis indicates that all of the methods assume that stakeholders, at least tacitly, know the importance of requirements, or that estimation methods are available for practitioners to use.

The following tables present examples of the studies conducted on requirements priorization methods. Research settings, goals and results of the studies are summarized in the tables. Table 6 introduces studies in which product developers prioritized the requirements; Table 7 focuses on studies in which university students prioritized the requirements; and Table 8 examines studies in which the authors (of the articles) prioritized the requirements.

Table 6. Examples of studies where product developers prioritized the requirements

Techniques	Subjects	Requirements	Goals	Results
Pair-wise comparison technique (cost-value approach) *versus* numerical assignments (Karlsson 1996)	5 participants from Ericsson Radio Systems AB	14 candidate requirements	Compare methods according to fastness, results' informativeness, trustworthiness, subjective response, positive side effects, and which technique to recommend.	- AHP is useful and efficient in practice. - AHP determines more precisely the importance and cost distribution of the requirements.
Cost-value approach (Karlsson et al. 1997b)	Ericsson Radio Systems AB, RAN-project,	14 candicate reqs that were already implemented or declined 11 high-level functional reqs suggested for the fourth release of a project	Apply cost-value approach in two projects	- Cost–value approach is useful for prioritizing requirements - Cost–value diagrams let management take action
Improved cost-value approach (Karlsson et al. 1997a)	Developers from Ericsson Radio Systems AB	23 reqs dealing with product improvement, evaluation of the process and the usefulness of the process and its tool support	Evaluate improved cost-value approach in an industrial project.	- Implementing global and local stopping rules that reduced the required number of pair-wise comparisons 75% - Introduces an initial strategy for dealing with requirements interdependencies.
Distributed prioritization process (Regnell et al. 2001)	10 stakeholders from Telelocig AB	58 features. One prioritization for individual features, other for feature groups.	An industrial case study where a distributed prioritisation process is proposed, observed and evaluated.	- Distributed prioritisation is useful, and the visualisation charts are valuable decision support. Challenges: - Difficulties with absolute assessment - Assessment of prioritisation quality - Sensitivity to 'shrewd tactics
Release planner (Carlshamre 2002)	Cooperative evaluation with planning experts from three companies	Excerpts from the companies' actual requirements databases_ Case A: 74 reqs Case B: 200 reqs, 47 of which analysed Case C: 1100 reqs, of which 172 were candidates for the release	Design, implement and evaluate a support tool for release planning as a means to provoke a rich understanding of the task of release planning	- Release planning is a wicked problem - The value of a release is hard to define - Criteria cannot be defined in advance - Judgements are relative - Planners discover properties as they plan - Need support for alternative models - One release is not enough - Credibility issues
EVOLVE (Greer et al. 2004)	Sample project	20 requirements	-	-
Multi-Criteria Preliminary Requirements Ranking Technique (MCPRRT) (Sobczak et al. 2007)	5 leaders of a city in Poland	11 requirements	-	- MCPRRT process helps in systematic analysis, but is time consuming and laborious. MCPRRT takes viewpoints into account, but leads to single list.

AHP *versus* CBRank (Perini et al. 2009)	23 Ph.D students and Junior Researchers	20 requirements from a real project	Compare methods according to the ease of use, the time-consumption and the accuracy	- For the first two characteristics CBRank overcomes AHP - For the accuracy AHP performs better than CBRank - The majority of the users found CBRank best
Pilot the Joint release planning method (Heikkilä et al. 2010b)	140 stakeholders, including over 10 development teams	Real requirements of different development teams	- Improve coordination of work of multiple agile development teams who develop a large legacy software product - improve communication between product management and development	- Joint release planning method is a significant advancement to find ways to scale agile software development methods to multi-team software devel-opment contexts

Table 7. Examples of studies where university students prioritized requirements

Techniques	Subjects	Requirements	Goals	Results
Analytical hierarchy process (AHP) and variation of the Planning Game (PG) (Karlsson et al. 2004)	15 PhD students and one professor	Mobile phone features (8 and 16)	Compare methods according to: - ease of use, - the time-consumption, and - the accuracy	- Straightforward and intuitive PG was less time consuming, and considered by the subjects as easier to use, and more accurate than AHP.
Tool-supported pair-wise comparisons versus planning game (Karlsson et al. 2007b)	15 PhD students and one professor	Mobile phone features (8 and 16)	Compare methods according to: - ease of use, - the time-consumption, and - the accuracy	- Tool-supported pair-wise comparisons is faster and as easy to use as the Planning game. The techniques do not differ significantly regarding accuracy.
Two ways of calculating priorities in hierarchies (with HVC) (Berander et al. 2009)	18 Master students	Features for the next generation course management system (6 feature groups having 1-7 lower level requirements, 21 in total)	- Empirical evaluation with respect to accuracy of the results - Determine whether to use compensation factor of not when prioritising unbalanced hierarchies.	- The amount of information of the req hierarchy does not direct the users to prefer a compensated or an uncompensated priority calculation. - Priority calculation that compensates for block sizes is in preferred over an uncompensated. - Subjects are pleased with HCV, and believe that it is scalable.
AHP *versus* CBRank (Perini et al. 2009)	23 PhD students and Junior Researchers	20 requirements from a real project	Compare methods according to: - ease of use, - the time-consumption, and - the accuracy	- CBRank overcomes AHP in ease of use and time-consumption - For the accuracy AHP performs better than CBRank - Majority of the users found CBRank best
Numerical Assignment (NA), Analytic Hierarchy Process (AHP) *versus* Extensive Numerical Assignment (ENA) (Voola&Babu) ((Voola et al. 2013)	Group of students	15 requirements	Examine the three techniques using various objective and subjective measures like number of decisions, time consumption, ease of use, attractiveness, scalability and reprioritizability	- ENA transcends NA and AHP

Table 8. Examples of studies where authors of the paper prioritized requirements

Techniques	**Subjects**	**Requirements**	**Goals**	**Results**
AHP, hierarchy AHP, spanning tree matrix, bubble sort, binary search tree, priority groups (Karlsson et al. 1998)	3 authors of the paper	“13 well defined quality requirements on a small telephony system” (generic quality requirements)	Illustrate, evaluate and characterize the methods	AHP is demanding, but worth the effort because its ability to provide reliable results and support knowledge transfer and consensus reaching
AHP and Cumulative Voting (CV) + Cumulative voting (CV) (4 cases) versus Hierarchical Cumulative Voting (HVC) (2 cases) (Berander et al. 2006a)	17-19 respondents in each case	14-25 objects such as decision criteria or process improvement issues	Compare strengths and weaknesses of AHP and CV. Empirical results from HVC.	HCV seems very promising
100$ Dollar method and Binary Search *versus* modified 100$ Dollar method and Binary Search	2 authors of the paper	7 short requirements	– analyze the variations among the results of these Requirements Prioritization Techniques – discuss attributes that can affect the requirements prioritization when dealing with Geographically Distributed Stakeholders	Prioritization techniques must be modified to fit geographically distributed teams. (The paper does not discuss how this should be done.)

2.5 Short summary of the literature review

Previous sections introduced the key areas, concepts and earlier research releated to our research. This section draws together and summarizes the main points from the existing liteterature in these different areas.

A starting point for our research is that the ultimate goal of the software product development organizations is to provide value (e.g. (Boehm et al. 2003). Traditionally, companies have tried to ensure value for their customers by selecting the most valuable requirements to be implemented in their products (e.g. (Gorchels 2000; Penny 2002a)).

However, requirements prioritization has been reported as a difficult activity in practice (e.g. (Carlshamre 2002)). Companies do not know how requirements should be prioritized so that the decisions would provide value. Even the concept of value is somewhat volatile (Racheva et al. 2009). It has been difficult to precisely define what value actually is and what its incredients are. In general, estimating and calculating software value is difficult (Khurum et al. 2013).

Long-term product planning (roadmapping) is a higher-level attempt to select features to be implemented in the software products and thus provide value. Although, the origins of roadmapping as a technique are in the traditional industry (e.g. (Kappel 2001; Phaal et al. 2003)), there are studies about roadmapping in software context (e.g. (Regnell et al. 2008; van de Weerd et al. 2006; Vähäniitty et al. 2009)). However, business research investigates value constructs from a number of perspectives, but in the software engineering literature, these have little or no explicit connection to software product planning and development (Khurum et al. 2013).

The research on requirements prioritization has been focused mostly on prioritization techniques and their evaluation (Pergher et al. 2013). The literature offers techniques for requirements prioritization (e.g. (Boehm et al. 1998; Karlsson et al. 1997b; Wiegers 2003), but there is no thorough analysis of the suitability of approaches for solving practical challenges. The current state of requirements prioritization and long-term product planning in market-driven software product development also do not seem to have been widely investigated. This means that the problem area is somewhat fuzzy, which may make developing solutions for it difficult. Furthermore, the connection of requirements prioritization and long-term product planning to customer value creation in practice needs further investigation.

Based on all this, the aim of this thesis is to investigate the current state of requirements prioritization and long-term product planning, as well as their linkages to customer value creation in market-driven software product development.

3. Research methodology

The aim of this thesis is to investigate the current state of requirements prioritization and long-term product planning and their linkages to customer-value creation in market-driven software product development. We wanted to gain experience and construct a thorough picture of the situation in which practitioners do long-term product planning and prioritize requirements. Thus, we selected a qualitative research strategy. The work was conducted in real product development organizations.

This section first gives an overview of the research project and the case organizations in which the research work was conducted. Furthermore, it describes the research approach and methods used.

3.1 Research context and case organizations

This research was conducted in the Software Business and Engineering Institute (SoberIT) at Aalto University School of Science (earlier Helsinki University of Technology (HUT)) as part of the QURE (Quality Through Requirements), CORE (Competitive Advantage through Stakeholder-Driven Requirements Engineering) and Reflex (Providing Value with Flexible Requirements Engineering) research projects. All of the projects were financed by Tekes (The Finnish Technology Fund) and by the companies participating in the project.

3.1.1 The QURE, CORE and Reflex projects

The goal of the QURE research project was stated as follows: how can organizations cost-effectively develop products that better satisfy users' and customers' needs? The goals of the research were to develop requirements engineering models, methods, and practices and test them with the industrial partners. The QURE project lasted from the year 1999 to the year 2002.

The goal of the CORE research project was to develop systematic practices for Finnish software development organizations so that they can cost-effectively involve stakeholders in developing products that satisfy customer and user needs. In order to reach the goal, the project developed and adapted a set of critical processes, methods, and tools, and supported their transfer to practice in the participating industrial partners. The CORE project lasted from the year 2003 to the year 2006.

The goal of the Reflex research project was to investigate how companies can flexibly and cost-effectively develop whole products that provide value for both customers and users. In order to reach the goal, the project developed a set of critical models, methods, and practices, and supported their transfer to practice in the participating industry. The REflex project lasted from the year 2006 to the year 2009.

3.1.2 The case organizations

Data were collected between years 2000 and 2007 in total 7 Finnish software product development companies. The case organizations and their application domains are introduced in Table 9.

Table 9. Case organizations and their application domains

Company	Number of employees	Application domain	Product types
A	23000	Transportation systems for buildings	Embedded systems
B	1100	Measurement systems for meteorology, environmental sciences and traffic safety	Interactive systems
C	500	Information management systems for building, public infra and energy distribution designers	Software systems
D	300	Computer security systems for companies and consumers	Software systems
E	300	Systems for financial processes and buyer-supplier related transactions	Software systems
F	100	Computer security systems for companies and consumers	Software systems
G	25	Web-aided change consultancy systems	Software systems

3.2 Qualitative research strategy

The aim of this thesis was to investigate the current state of long-term product planning and requirements prioritization and their linkages to customer value creation in market-driven software product development. The need for a deep understanding of long-term product planning and requirements prioritization practices, problems and needs in software product development companies called for a qualitative research strategy. According to Avison et al. (1999), the particular strength of qualitative methods is their value in explaining what goes on in organizations. In addition, qualitative methods permit the evaluator to study selected issues in depth and detail (Patton 2002).

Since our focus was on understanding the complexity of issues involving human behaviour (Seaman 1999), we decided to follow a qualitative research strategy exclusively. This means that we did not have any quantitative research methods in use. As a research method, we selected case studies, which are described in more detail in next section. Within individual case studies, we combined a variety of qualitative data collection techniques in real life organizations to understand the phenomena at hand deeply.

Our research described reality through the lenses of the practitioners. This means that our approach to the research was interpretative. In our setting, the independent and dependent variables were not predefined, as in a positivistic approach. Instead the objective was to understand the deeper structure of the phenomenon (cf. Pare, 2001). According to Myers (2009), positivist researchers generally assume that reality is objectively given and can be described by measurable properties, which are independent of the observer and his or her instruments. Interpretative researchers, instead, assume that access to reality is only through social constructions (Myers 2009). Interpretive studies assume that people create and associate their own subjective and intersubjective meanings as they interact with the world around them. Interpretive researchers thus attempt to understand phenomena through accessing the meanings that participants assign to them (Boland 1985; Orlikowski et al. 1991).

3.3 Case study approach

We wanted to gain a representative view of how practitioners do requirements prioritization and long-term planning, what challenges they have, how requirements prioritization methods solve these practical challenges and how these activities relate to value creation. Therefore, we designed four case studies that were supposed to show the different aspects of the phenomena.

A case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between the phenomenon and the context are not clearly evident (Yin 1994; Yin 2009). One of the major strengths of the case study method is the opportunity to use many sources of evidence (data triangulation) and many data collection methods (methodological triangulation) (Yin 1994; Yin 2009).

We used an 'industry-as-laboratory' research approach as Potts (1993) suggests, in which researchers identify problems through close involvement with industrial projects and create and evaluate practices addressing those problems. This lets researchers emphasize what people actually do or can do in practice, rather than what is possible in principle. Formal research operates at a distance from the practitionersä everyday lives and, although it provides interesting theoretical perspectives about the nature and complexities of social life, it largely fails to penetrate the experienced reality of their day-to-day work (Stringer 1999).

Our research method could be also characterized as action research. We also intended to improve current practices with the organizations, and the cases were not "purely observational as case studies" (Runeson et al. 2009). In our cases, the researchers and the practitioners collaborated to solve the problems at hand, which led to reflective learning. The phenomena under study already seemed like such a complex real-life problem, that it might not be possible to investigate that problem deeply from the distance. Since Runeson et al. (2012) basically suggest applying the same guidelines for action research and for case studies, we designed our research as case studies even though they had an action research component.

Action research with its purpose to “influence or change some aspect of whatever is the focus of research” (Robson 2002) is closely related to the case study. For example, Runeson et al. (2012) prefer including action research in the wider notion of the case study, and for research they apply the same guidelines. This means that action research (Stringer 1999) can, in a way, be seen as a special type of case study (Runeson et al. 2009). Action research can address both complex real-life problems and immediate concerns (Avison et al. 1999).

In action research, the client organization and the researchers collaborate to solve a practical problem while also contributing to research. The researcher becomes a facilitator or consultant who acts as a catalyst to assist stakeholders in defining their problems clearly and to support them as they work toward effective solutions to the issue that concerns them (Stringer 1999). His or her role, in this context, becomes more facilitative and less directive (Stringer 1999). Action research seeks to engage ’subjects’ as equal and full participants in the research process (Stringer 1999). In our research, this meant that the researchers were, in addition to their investigative role, active members of the improvement projects, in which the existing practices were improved or prioritization methods were evaluated.

According to Avison et al. (1999), action research is an iterative process involving researchers and practitioners acting together in a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning. According to Stringer (1999), the basic action research routine consists of three iterative phases: look, think, and act. As participants work through each of the major stages, they explore the details of their activities through a constant process of observation, reflection, and action (Stringer 1999). In our research, this meant that the researchers presented summaries and analyses for practitioners during the research process, which guided how to continue.

3.4 Overview of research cases, data collection and data analysis

In this section, the study design is introduced in more detail. First we introduce the research cases, then we provide an overview of the data collection techniques used and finally we describe the data analysis techniques used.

3.4.1 Research cases

Studying multiple cases makes it possible to build a logical chain of evidence (Yin 2009). Our study was designed to consist of four individual research cases each of which would answer to one research question. The four cases are individual in the sense that they investigated different aspects of the phenomenon.

Even though the cases were individual, their research designs were partly built on each other. The designs of the later case studies were partly built on the earlier findings, which means that they were refined based on the findings from earlier cases. This means that the results of the earlier cases informed the design of the latter cases. The earlier cases had an effect on the goals of the studies and on the questions asked in the interviews[1].

The four cases of the study are presented below:

The goal of the Case 1 was to *understand the suitability and characteristics of requirements prioritization methods*. It consisted of an evaluation of two requirements prioritization methods and requirements prioritization process improvement in case organization A. This case was explanatory.

The goal of the Case 2 was to *clarify the current practice and practical challenges in the requirements prioritization area*. It consisted of a focus group study in case organizations B and C, and 12 state-of-practice interviews in all case organizations focusing on requirements prioritization challenges and practices. This case was exploratory.

The goal of the Case 3 was to *clarify the current practice and practical challenges in the long-term product planning*. It was a longitudinal case study consisting of in total 17 state-of-the-practice interviews in all case organizations, and it examined related long-term product and solution planning process improvement in case organizations C and D. This case was exploratory.

The goal of the Case 4 was to *identify and develop practices that support solution planning and development*. It was a case study consisting of 8 interviews and a set of workshops about solution planning and customer value creation, with in total 10 participants in case organization D. This case was exploratory.

Furthermore, all the research cases incorporate data collection via document analysis, informal conversations, experience-exchange seminars and observation in the case organizations during the whole research period. The summary of the research cases is introduced in Figure 3.

[1] The clearest example of this building upon an earlier case is the Case 1, based on which we realized that we could not continue our research towards developing our own requirements prioritization method. First, the whole phenomenon in the market-driven software development context must be better understood and analysed.

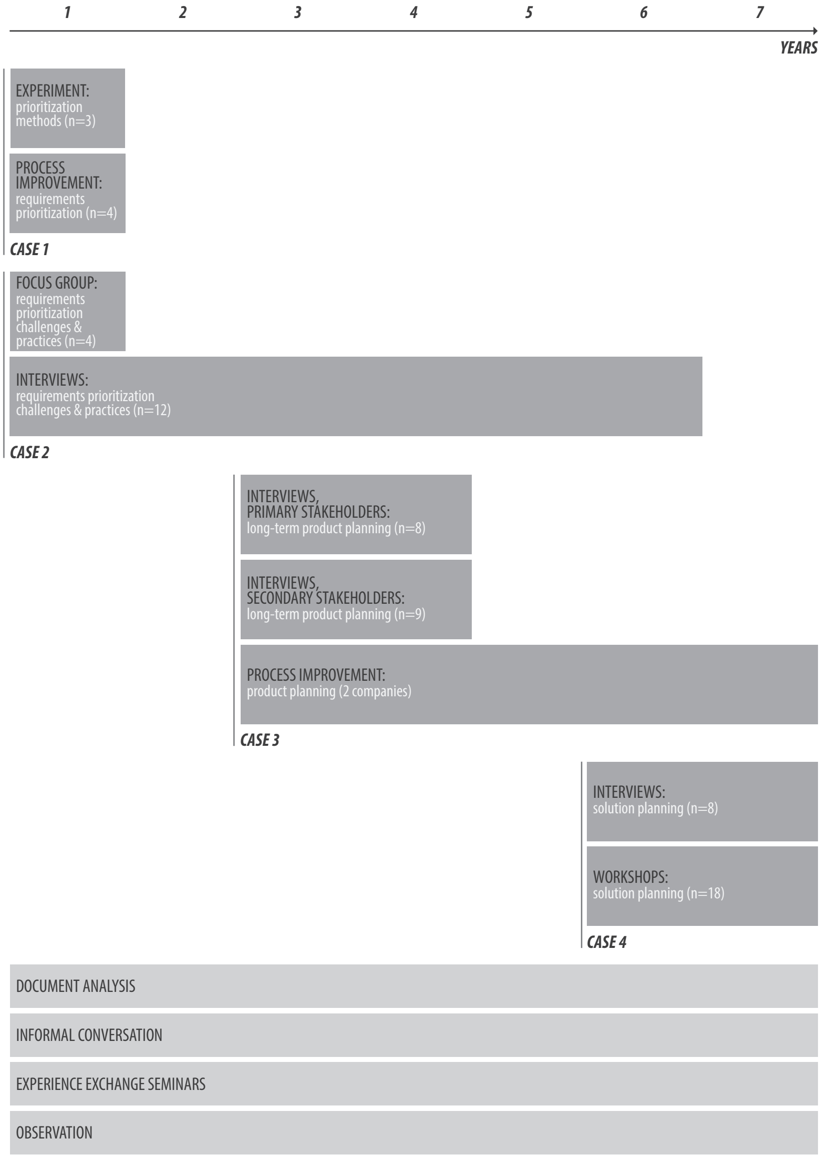

1
2
3
4
5
6
7
YEARS
EXPERIMENT: prioritization methods (n=3)
PROCESS IMPROVEMENT: requirements prioritization (n=4)
CASE 1
FOCUS GROUP: requirements prioritization challenges & practices (n=4)
INTERVIEWS: requirements prioritization challenges & practices (n=12)
CASE 2
INTERVIEWS, PRIMARY STAKEHOLDERS: long-term product planning (n=8)
INTERVIEWS, SECONDARY STAKEHOLDERS: long-term product planning (n=9)
PROCESS IMPROVEMENT: product planning (2 companies)
CASE 3
INTERVIEWS: solution planning (n=8)
WORKSHOPS: solution planning (n=18)
CASE 4
DOCUMENT ANALYSIS
INFORMAL CONVERSATION
EXPERIENCE EXCHANGE SEMINARS
OBSERVATION

Figure 3. Summary of the research cases (n= number of informants)

3.4.2 Overview of data collection

We wanted to gain a representative view of the suitability of prioritization methods, the current practices and challenges of requirements prioritization and long-term planning, and their role in value creation. Therefore the research design combines several data collection techniques. The benefit of using many data collection techniques within a case study is methodological triangulation. Interviews before workshops and process improvement work made it possible to try to solve the correct problems with the practitioners, as the problems were understood correctly before the actions were planned.

Interviews of the practitioners formed the basis of our data collection. We interviewed the practitioners individually, in pairs and in a group. We also observed practitioners performing their actual work and participated in companies' process improvement work in these areas. In addition, we organized experience exchange seminars in which the practitioners presented their practices and interacted with practitioners from other companies. Furthermore, we analysed different kinds of templates, documents and materials that companies produced related to the subject.

The motivation for using these different kinds of qualitative data collection techniques was to get a multidimensional picture of the phenomena. We wanted to know what practitioners do in practice, how they describe their work in the area of requirements prioritization and long-term planning, how they discuss this work with the other practitioners, what kind of templates they have for planning and how those templates are used in practice. We also wanted to follow how improvement actions actually help solve the existing challenges over a long period of time.

The data collection techniques we used are listed in Table 10 and the purposes of each study activity are summarized in more detail below. In Table 10, 'x' means that the research activity was performed in the company, and '(x)' means that prioritization methods were evaluated in the company, but not that formally.

Table 10. High-level view of research activities

	A	B	C	D	E	F	G
Individual and pair Interviews		x	x	x	x	x	x
Focus group	x						
Document analysis	x		x	x	x	x	x
Experience exchange seminars			x	x	x	x	
Informal conversation	x	x	x	x	x	x	x
Process improvement work	x		x	x			
Observation			x	x			x
Evaluation of prioritization methods	x						(x)

Interviews

Semi-structured interviews were one of the main data collection techniques used throughout the study. By semi-structured interviews, we collected data directly from the practitioners who prioritized requirements and did long-term product planning in the case companies; who were affected by these decisions or who affected these decisions. The motivation for selecting semi-structured interviews as a data collection technique was a clear choice in order to cost-effectively get lots of information about the phenomena using practitioner's own vocabulary.

We interviewed individual practitioners for 1 to 2 hours. Most of the interviews were tape-recorded. A couple of the interviews were pair interviews. In most cases, we had two researchers participating in the interview. One of the researchers asked questions and the other researcher made notes. However, the researcher making notes was capable of monitoring the situation and asking additional questions if needed. The purpose of this researcher triangulation was to increase the validity of the research.

Focus group

A focus group was used to obtain a wider insight on the practices used in different phases of development work and to get practitioners with different roles to interact with each other. For a focus group, individuals are selected and assembled by researchers to discuss and comment on, from personal experience, the topic that is the subject of the research (Powell et al. 1996). A focus group is a special type of group interview. We used this method with four practitioners from two case companies. The details of this data collection technique are described in in Article I.

Observation

To get hands-on understanding about the current practices and challenges, we observed requirements prioritization and long-term planning work in different phases of product development in companies C and D, in which we were also part of the process improvement work regarding the issues.

Experience exchange seminars

We organized experience exchange seminars concerning requirements prioritization, long-term product planning, solution planning and customer value creation amongst the practitioners in participating companies. In these seminars practitioners presented their current processes and challenges in the requirements prioritization, long-term product planning or solution planning areas and they were able to get feedback from other practitioners and from the researchers. The researchers were able to both collect the presented material and analyse the interaction between the participants (e.g. the themes that they wanted to discuss most).

Informal conversations

We had informal conversations about requirements prioritization, long-term product planning and solution planning with practitioners from all case organizations during the years of cooperation. Informal conversations took place in lunch discussions during seminars, over phone calls or in planning meetings with the practitioners from participating companies. Findings from these conversations were written down in the research diary kept by the researcher.

Document analysis

We collected and analysed the process documentation that practitioners from case organizations provided regarding requirements prioritization and long-term product or solution planning. The documents included requirements documents, roadmaps, long-term business plans, process descriptions, process charts, product development project plans and process improvement project plans. We asked practitioners to provide templates and examples of how the documents were used. In addition, all the documents or artifacts that the practitioners mentioned in the interviews were asked from the practitioners.

From the documents, we analysed which issues are planned in which documents and compared different companies' documents to find similarities and discrepancies. We also analysed which issues were not explicitly planned for in any of the documents.

Cooperation and process improvement in the whole RE area
We cooperated closely and became involved in RE process improvement and actual product development work with in total 7 companies. In each company, we studied RE activities for at least for four years during the period from 1999 to 2008. We also attended numerous meetings and process improvement activities that were not directly connected or supposed to be connected to requirements prioritization or long-term product planning, but which provided us with important additional contextual data. This extra information is based on the data collected through formal semi-structured interviews, observation, informal conversations, and analysis of R&D process documentation, and requirements specifications.

3.4.3 Overview of data analysis techniques

The data analysis techniques used in the study are summarized in this section. The detailed data analysis of different cases is discussed along with each case in Sections 3.5, 3.6, 3.7 and 3.8.

According to Myers (2009) a clear distinction between data gathering and analysis in qualitative research is problematic, as the analysis will affect the data and the data will affect the analysis and vice-versa. Furthermore, the data analysis in this study was iterative and progressive by nature, which also means that the data collection and analysis were partly simultaneous.

Since the cases had their own research questions, the data was analysed by within-case analysis (Eisenhardt 1989), and we did not do a cross-case analysis. However, the cases were formed so that we studied several companies within individual cases (Cases 2 and 3). In those cases, we had the possibility to search similarities and differences of the companies.

Coding and categorizing are two processes commonly adopted in analysing when beginning to make sense of the data (Simons 2009). These two processes formed also the basis of our data analysis of the interviews.

In cases 2, 3 and 4 we used for data analysis a procedure introduced in Simons (2009). We first organized the raw data from interviews, and used that data to identify a preliminary framework (which was originally created on the basis of themes from our interviews). After that, we organized data into a framework, used that framework for descriptive analysis and then did a second order analysis. This kind of iterative approach matches with Stringers (1999) basic action research routine which consists of three iterative phases: look, think, and act.

Preliminary analysis of the already-collected data was an elementary part of the following data collection. After the first interviews in each case, we redefined the following interview outlines and asked the practitioners about the issues that had arisen in earlier interviews.

In practice, the transcribed interviews were coded according to the process introduced above to analyse the data from different angles. This led to an understanding of, for example, the common challenges that practitioners experience. Furthermore, we tried to find similarities and discrepancies between companies and between different roles.

In Case 1, the data analysis was not as iterative as in other cases. We categorized the findings from the interviews and meetings, in which the prioritization techniques were experimented, basically from the viewpoints of benefits and challenges in using the methods. In addition, we searched for similarities and discrepancies from using the different methods in order to find common denominators.

The field notes, case write-ups and documents have a supportive role in the data analysis. They were not color-coded, but we, for example, compared the roadmaps that different companies produced. The goal was to analyse the issues they planned in product level, the issues that they planned in business level and the issues are not at all visible in the documents, even though they are mentioned in the interviews as planned items.

3.5 Case 1: Evaluation of requirements prioritization methods

3.5.1 Process improvement group

The starting point for the whole study was the recognized need in the case organization A to improve its requirements prioritization practices. In case organization A, an improvement group was established in the organization. The goals for this temporary group were to find out a suitable requirements prioritization method from the literature, to evaluate it, and to introduce and adopt it in the product development organization. The group consisted of a usability expert, a visual designer, two project managers, and an external researcher who worked as the facilitator of the group.

The practical role of the researcher was that of a facilitator. In the process improvement group, the researcher made notes and helped the group to form a direction. She, for example, summarised issues and presented these findings to the group along the way.

3.5.2 Evaluation of requirements prioritization methods

To understand the suitability and characteristics of requirements prioritization methods, we evaluated two requirements prioritization methods from the literature, the pair-wise comparison technique (Karlsson 1996) and Wiegers' method (Wiegers 1999), in the case organization. The two case projects were selected on the basis of their interest in adopting a prioritization method. Both of the projects decided to try the prioritization method that they thought would be the most suitable in their case. Both of the studies were performed in real product development projects that had different challenges.

In both cases, we selected relevant participants within or outside the project and a subset of the project's requirements (see Table 11). The participants prioritized the requirements with the prioritization method according to the instructions given in the literature. However, the project managers were given the opportunity to make small adjustments to the methods if they felt that something in the method would not be suitable in their case.

The role of the researcher was to provide help in using of the methods and to collect the experiences by making notes and having discussions with the participants. We also collected the practitioners' experiences and attitudes towards the prioritization methods through a questionnaire.

Table 11. Case projects and requirements

	Type of requirements	Evaluators	Prioritization method
Project Alpha	User needs	4 users of the system	Pair-wise comparison technique
Project Beta	Change requests	Project manager Requirements engineerg	Wiegers' method

3.5.3 Data analysis

To analyse the data, we categorized the findings from the prioritization sessions based on the experienced benefits and disadvantages experienced when using the methods. The findings from the sessions were compared to each other in order to find similarities, which can be seen as searching for cross-case patterns (Dubé et al. 2003). We did not compare the methods with each others, but instead tried to find common benefits and challenges in using prioritization methods.

This part of the research was mostly done in work meetings, which were neither recorded nor transcribed. However, the researcher made field notes along the way and validated her findings by discussing them informally with practitioners after the prioritization sessions. The questionnaire also provided important additional data about practical experiences and attitudes towards the prioritization methods. The data gathered from the process improvement group meetings provided additional data and supported the findings of the method evaluation part.

3.6 Case 2: Requirements prioritization challenges and practices

To clarify the current practice and practical challenges in the requirements prioritization area we carried out a focus group study between participants from two case organizations (B and C) and interviewed in total 12 practitioners in seven case organizations A-G. In addition, we held experience exchange seminars between companies, had informal conversations with practitioners, observed requirements prioritization in practice in two companies and analysed related documents in six companies.

3.6.1 Focus group

The goal of the focus group session was to find out how and in which phases of the development work companies prioritize requirements in practice. We also clarified which aspects affect priorities and from which sources the practitioners gather the information on which they base their priority decisions. In addition, we inspected the problems the developers felt they had with their current requirements prioritization practices.

The researchers selected a wide range of practitioners for focus group and individual interviews in order to obtain wider insight of the practices in use in different phases of the development work. The interviewees represented different roles in the companies. Participants of the focus group represented roles from both product management and project management (see Table 12).

Table 12. Focus group participants

Company	Role
B	Project manager
B	Product development process engineer
C	Product manager
C	Leader of R&D unit

3.6.2 Individual interviews

The goal of the individual interviews was to get information about the current requirements prioritization practices in the software product development companies and about the models that practitioners base their priority decisions at the moment. There were in total 12 interviewees in 7 case organizations.

The distribution of the different roles in individual interview was 5 project managers, 2 software architects, 4 product managers or persons in charge of product management and 1 usability expert (see Table 13).

Table 13. Case 2 interviewees

Company	Title	Viewpoint	Interview type
A	Usability expert	Project	Individual
B	Project manager	Project	Individual
C	Director of product management	Product management	Individual
C	Project manager	Project	Individual
D	Project manager	Project	Pair interview
D	Quality manager	Project	Pair interview
E	Product manager	Product management	Individual
E	Software architect	Project	Individual
E	Software architect	Project	Individual
F	Director of product development	Product management	Individual
G	Project manager	Project	Pair interview
G	Product manager	Product management	Pair interview

3.6.3 Data analysis

All of the interviews except one of the pair interviews were transcribed from the recordings. Soon after the interviews, the most significant findings from each interview were organized into mind maps for further processing. The mind maps were completed using the transcriptions, and the opinions of the interviewees were compared to find similarities and discrepancies. In addition to comparing all of the interviewees at the same time, we made role-based comparisons to understand role-specific characteristics in the experiences and challenges involved.

During the improvement work in the two case organizations, the researcher kept field notes in a research diary. Case write-ups were created after every meeting. The field notes were mostly descriptions (each about a half-page long) of what happened in each meeting and, typically, a few sentences of analysis of observations concerning, for example, conflicts between participants or participants' motivation. The field notes can be seen both as notes from the meetings and as a type of preliminary analysis.

3.7 Case 3: Long-term product planning challenges and practices

To clarify the current practice and practical challenges in the long-term product planning, representatives from different stakeholder groups were interviewed. In practice, we interviewed primary informants from four companies and secondary informants from two case organizations. By this, we wanted to understand the current long-term product and solution planning processes from different viewpoints. We were also involved in process improvement in the long-term planning area in two case organizations. Furthermore, we held 8 interviews and a set of process improvement workshop in the area of solution planning in one case organization.

3.7.1 Interviews of primary and secondary informants

In total 8 primary informants were interviewed in four case organizations C, D, E and F. By primary informants we mean practitioners who are personally responsible for preparing long-term product plans. In addition, we interviewed 9 secondary informants in the two main case organizations C and D. By secondary informants we mean practitioners that have a business interest in long-term plans. The key criterion for selecting the primary informants was that they have experience and knowledge about the current planning activities. Regarding secondary informants, it was important that the interviewees represented different perspectives, such as marketing, services, documentation, and R&D. Informants are introduced in Table 14 and Table 15.

Table 14. Primary informants

Company	Informant
C	Product managers (n=2) Product management director
D	Product managers (n=3)
E	Product manager
F	Director of product development

Table 15. Secondary informants

Company	Informant
C	Business area leader Director of a customer segment Documentation specialist Marketing planner Service manager
D	Director, professional services Director, services Director, software processes Vice president, R&D

The interviews were semi-structured, meaning that there were certain themes that we discussed in each interview, but the phrasing of the questions varied from time to time. The interview themes included current long-term planning practices, challenges and good practices, and future development ideas that the practitioners have for improving their current practices. The interviewees were also asked to show and explain any documents, process descriptions, and examples related to long-term product planning that they had in the company. The researchers were also able to take this material with them in order to perform a deeper analysis. The interviews lasted from 1 to 2 h. All of the interviews were recorded.

3.7.2 Process improvement concerning long-term product planning

In addition to the interviews, the researchers were involved in process improvement work in case organizations C and D, which allowed them to attain a deeper understanding of the existing processes and practices, improvement needs, and viewpoints of different stakeholders. The researchers were also able to observe actual long-term planning sessions in the two main case organizations C and D.

We also organized three experience exchange seminars for the practitioners of all of the case organizations. In these seminars, the practitioners presented their long-term planning practices and were thus able to learn from each other and gain new understanding and insights. The researchers, on the other hand, had an opportunity to follow the current trends and learning curve in the companies. In practice, the researchers worked closely with the two main companies in order to improve their long-term planning practices in co-operation with the practitioners during the research period.

The practical role of researchers during the improvement process was that of a kind of facilitator; the practitioners asked for their advice and comments for improving the process, as well as had spontaneous discussions about the current challenges of the organization. They also gave comments on early drafts of the process and templates as well as interviewed practitioners other than the ones responsible for the process development in order to learn different opinions about the processes.

3.7.3 Data analysis

All of the interviews were transcribed from the recordings. The analysis of the interview data was performed in three stages. The data were analyzed first on an interviewee-by-interviewee basis, then within each case organization, and finally all together. Quotations related to each theme discussed in the interviews were grouped together. Groups related to different themes were represented using a mind map.

In the first phase, this grouping of quotations was separately sorted for each interview. In the second phase, the mind maps created in the first phase were combined so that the findings from all the interviews in one company were put into one mind map. In this phase, the researchers also started to build up thematic categories on the basis of similar quotations from different interviews. This resulted in a number of sub-branches of common findings within high-level theme branches in the mind maps. In the third phase, the mind maps were combined so that the findings from all the interviews were put into a single mind map. The researcher was able to find more categories by recognizing common findings from the interview quotations.

The focus of the study was more on finding commonalities amongst the companies in their long-term planning, rather than finding discrepancies and analysing the rationales for these discrepancies. However, we also made role-based comparisons to understand the role-specific characteristics in the experiences and challenges involved in long-term planning within the case organizations.

The interview results were validated in case organizations C and D. The researchers started the process improvement work in these case organizations by presenting the interview results to the practitioners. In both organizations, these findings were discussed with a representative group of practitioners. These discussions provided the researchers with new insights that helped them understand the interview findings better. Within these validation discussions the misunderstandings from the interviews were also corrected.

During the improvement work itself, the researcher kept a research diary in which she wrote her field notes. Case write-ups were written soon after every meeting. The field notes were mostly descriptions about half a page long of what happened in each meeting. They typically included a few sentences of analysis of observations concerning, for example, conflicts between participants or the motivation of participants. The field note findings were not added as such to the mind maps, but they were compared to the findings from the mind maps. Even though only one researcher did the interviews in this part of the study, another researcher, who also participated in the process improvement, reviewed the findings.

The results described two main areas: the current state of the case organizations' planning activities and the practices that seem to link business to RE. The description of the current state was formed on the basis of commonalities between the companies. The practices that link business to RE were selected on the basis of the researchers' analysis of the current practices and existing bottlenecks in the case organizations.

3.8 Case 4: Solution planning and customer value creation

The core of the further research collaboration with case organization D consisted of eight interviews and four workshops. The number of interviewees was eight, and ten different practitioners participated in the workshops.

3.8.1 Interviews

The goal of the eight interviews was to investigate the current state of solution planning in the company and to gain information about how the interviewees would improve the existing practices. The key criterion for selecting interviewees was that they had knowledge about the current planning activities. In addition, it was important that the informants represented different perspectives, such as strategy planning, long-term product planning, service planning, marketing, R&D, and process improvement.

The interviews were semi-structured. The researchers defined five topics for the interviews: (1) long-term planning activities; (2) customers and customer groups; (3) the benefits gained by customers; (4) the components of the solution, and (5) the RE activities and agile approach. In addition, the researchers defined a set of company-specific questions that specified these topics in more detail. The purpose of the topics and questions was to get the latest information about the planning and RE activities of the case organization.

3.8.2 Workshops

The results of the interviews were processed further in the workshops. The purpose of the workshops was to elaborate the interview results and to create a dialogue amongst the practitioners and between the practitioners and the researchers. The target was to identify and develop together practices that support solution planning and development.

The number of practitioners in the workshops varied from four to seven. Their role was to give feedback about the findings and suggestions presented by the researchers, make their own proposals, and provide additional knowledge about the current situation. At least two researchers participated in each workshop. Their role was to present the findings from the data gathered earlier, make suggestions, facilitate the discussion, make observations, and write notes. The last two workshops were also recorded.

3.8.3 Data analysis

All of the 8 interviews were transcribed from the recordings. Three researchers analysed and coded the transcripts. Each of them had his or her own themes for the coding. The themes were (1) long-term planning activities; (2) customers; (3) customer benefits; (4) the development of solutions, and (5) RE activities. During the coding phase, the researchers identified quotations related to the themes and marked these quotations with a colour code in the transcripts. The purpose of this color-coding was to enable the researchers to follow each other's coding process and to check that all of the relevant issues had been identified and marked.

After coding, each researcher analysed the coded material further from the perspective of his or her themes and grouped the quotations from the individual interviews into categories. For example, the analysis of the quotations related to long-term planning activities led to three main categories: (1) planning levels, (2) planning horizons, and (3) challenges related to planning.

Each researcher presented the results of his or her analysis to the other two researchers to validate the findings. On the basis of the analysis, the researchers wrote a case report that summarized the key findings from the interviews. The results of the interviews were validated and processed further dring the workshops that were organized for the practitioners of the case organization. In each workshop, the researchers presented the key findings from the previous workshop and introduced new ideas to support solution planning and development. These ideas emerged from the data gathered from the practitioners of company B, the existing literature, and the experience of the researchers.

As a summary, the data analysis of second phase was iterative. Two researchers analysed the collected data and evaluated the findings together before the workshops. In the workshops, the researchers and the organization's personnel elaborated on those findings.

3.9 Validity

The validity of a study denotes the trustworthiness of the results, and to what extent the results are not biased by the researcher's subjective point of view (Runeson et al. 2012). Defining the concept of validity unambiguously is difficult, as different authors have provided different definitions and placed emphasis on different aspects of the concept. Furthermore, there are different ways to classify aspects of validity and threats to validity. Some qualitative researchers even reject the framework of validity, for they reject the basic realist assumption that there is a reality external to our perception of it (For instance, Lincoln et al. (1985) propose a different framework for evaluating qualitative research.

In this thesis we use a classification theme used by Runeson et al. (2012). This selection of a classification theme may not be totally perfect because it has the backgound in quantitative research tradition, and the categories are not so easy to utilize in qualitative research. However, we wanted to use a well-known pattern for case studies in the area of software engineering and adapt those to our own interpretative case study.

According to Runeson et al. (2012):

- Construct validity reflects to what extent the operational measures that are studied really represent what is investigated according to the research questions.
- Internal validity is about causal relations and whether there is a risk that the investigated factor is affected by a third factor.
- External validity denotes the property of an empirical study where the result is generalizable to other contexts. However, since in case studies there is no population from which a statistically representative sample has been drawn, the intention is to enable analytical generalization.
- Reliability is about to what extent the data and the analysis are dependent on the specific researchers.

To increase the **construct validity** of the study, we decided to use several tactics, especially in the data collection phase. First, we triangulated data sources by selecting more than one informant from all case organizations. In addition, we selected interviewees so that different stakeholder groups within the companies were represented. This allowed us to investigate the phenomena from diverse perspectives. Second, we used the triangulation of data collection techniques to improve the construct validity of findings within the organizations. Our data collection techniques included interviewing, document analysis, informal discussions, and observation. In addition, we developed practices cooperatively with practitioners from the companies in order to gain deeper information about their current practices, challenges and attitudes. By using a multi-method strategy (Yin 2009) in the case study level, we were, e.g. able to relate the information gathered from the interviews to the observations. Third, the studies were carried on in the case organizations over a long period, which improved researchers' ability to understand the existing practices and practical challenges in more detail. Due to this prolonged involvement we were also able to validate the earlier findings later and make sure that we had understood findings right in the first phase.

The focus of our research was on understanding the phenomena, not on explaining causal relations. Therefore, the **internal validity** issues (according to definition used in Runeson et al. (2012)) were not the main concern of our research. However, even though our study was not about causal relations, there were elements in our findings that could be seen as somewhat causal explaining. E.g. our explanations for the causes of challenges in requirements prioritization could be seen as description of causal relations. Our actions to increase construct validity were in a way supposed to have an effect to internal validity as well. The different kinds of triangulations and the long research period were supposed to minimize the risk that some unexpected factor (e.g. one wrong interview question or a biased event) would have affected on our findings. In addition, we spent a sufficient time with the case organizations, which lowered the risk of that the results would be affected by a temporary or occasionally occurring factor.

To improve the **external validity** of the research results, this study involved seven separate case organizations of different sizes. In addition, we tried to select companies that represented different types of products, customers and business environments in order to understand the marked-driven companies more widely and to ensure that our explanations were plausible in market-driven settings. We also focused on collecting and analysing the data, and forming context descriptions in a way that we really understand what the “common characteristics” (Runeson et al. 2012) of the cases are.

To improve the **reliability** of the study, we paid attention to investigator bias by using investigator triangulation. On most occasions we had more than one researcher participating in interviews and other research activities. Furthermore, peer debriefing was in place also in the sense that all the interview outlines were either planned cooperatively or reviewed by at least one other researcher.

The compromises in our research design were mainly related to limited resources and the nature of the study. We had to select the companies and participants partly based on availability. We also had no chance to plan as many evaluation sessions as we wanted for evaluating the requirements prioritization methods.

4. Results

This section gives an overview of the contents of the included papers, and summarizes the main research results. The results are grouped in relation to the research questions stated in Section 1.2. Section 4.1 describes the state of the practice in long-term product planning and requirements prioritization. In Section 4.3, we show the practical challenges included. The suitability of requirements prioritization methods is described in Section 4.4. Section 4.5 introduces how to support the link from long-term planning and requirements prioritization to customer value creation.

4.1 State of the practice in long-term product planning and requirements prioritization

The first objective of this thesis was to investigate the current state of long-term product planning and requirements prioritization in market-driven software product development context.

This research objective has been addressed in four publications. Publications III and IV analyse the current state of long-term product planning. Publication II introduces practices that practitioners currently use to prioritize their requirements. Publication I is focused on practical challenges involved, but also sheds light on the state of the practice in requirements prioritization.

This section summarizes the current state of long-term product planning and requirements prioritization in practice.

4.2 Process and people involved

According to our findings, there were two main phases of product development where decisions about features to be implemented and their priorities were made. These phases are in this thesis called *product level* and *project level*. The product level refers to the phase before the projects are scoped in which the requirements are allocated to the forthcoming projects. Activities aiming at selecting features to be implemented were called long-term product planning, roadmapping, version planning and release planning in the case organizations. The project level refers to within-project activities. A very simplified picture about the levels and division between decision-making activites between them is summarized in Figure 4.

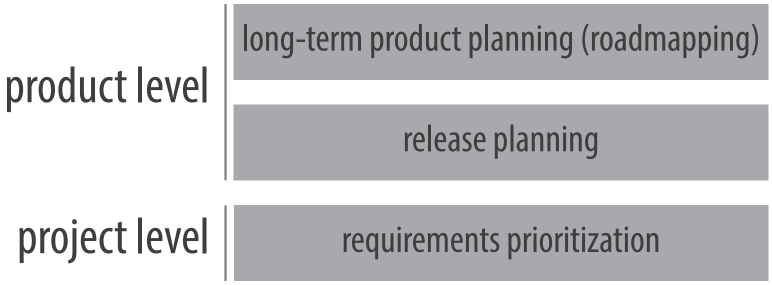

product level
long-term product planning (roadmapping)
release planning
project level
requirements prioritization

Figure 4. A simplified picture about the division of decision-making activities in product level and project level

On the *product level* the goal was to select the features to be implemented within products so that the maximum amount of value is provided to customers and users and thus to the company itself. The case organizations received a great number of ideas for potential features from the markets that were analysed and prioritized to certain product releases, typically called projects. In most of the case organizations these ideas were collected into an RE tool by different internal stakeholders. In the product level, the product managers were the main actors trying to allocate the manifold requirements gathered from the field for forthcoming releases of the products.

On the *project level*, within one project, the main goal of the prioritization was to organize the already-selected requirements into a sufficiently rational implementation order. The prioritization of requirements within a project usually consisted of many iterative prioritizations during the project. The main decision makers at this level in the case organizations were most often project managers who, in cooperation with product managers and software architects, allocated requirements to be implemented in the next iterations of an on-going project.

The first prioritization done in the beginning of the project was typically business based. In the case projects, the basis was usually the priorities nominated from the business viewpoint by product managers. However, as the projects evolved, technical issues started to play a bigger role and the implementation order of the requirements was constructed on the basis of technical issues. For example, during the more detailed definition of requirements the understanding concerning the size and potential risks of a requirement increased. One aspect that usually over-dominated the decisions within the projects was the maturity the definitions of different requirements at different points of the project.

4.2.1 Planning horizons

Our findings indicate that the focus of the planning is typically on one product in a short term at a time. This means that decisions concerning the requirements that will be implemented, as well as their implementation order were also made short term and for one product. The planning horizon in the case organizations was from one to two releases ahead. The time horizon for planning was usually kept open so that steps for the nearer future were planned in greater detail. The remote future was also outlined, but in less detail. The actual practices used to perform this kind of planning varied significantly and are introduced in more detail in Section 4.2.3.

Plans in different levels were not connected together in practice. In the case organizations, there were business plans for the business development, product roadmaps for describing the future of different products and technology roadmaps for technical development and possibilities in the future. The practitioners did not have practices for combining these plans together.

4.2.2 Planned items

The main content of roadmaps seemed to be the forthcoming features of individual products at the high level. In the case organizations, the roadmapping was usually done from the viewpoint of one product and the linking between products was not explicit. Practitioners longed for long-term product planning that would combine the future development steps of many products. However, both the practice (meetings, quarterly planning days etc.) and material (meaning: templates etc.) focused on individual software products and ability to understand the big picture of the products was difficult.

The practitioners in the case organizations needed also some articulated decisions - not only concerning the future features of subsequent releases, but also the expected customer segments, and which geographical areas they were to satisfy most with the different releases. This was seen as necessary for preparing for the future and for understanding the priorities of the potential features better. However, the practices and documents were not that established in that.

Furthermore, solutions as a whole were not explicitly planned and described. This means that what customers and users actually get, was not clearly described, nor was it explicitly decided where it was heading. In a few case organizations, the roadmap templates also covered issues such as services included, marketing arguments or product position in the markets, but practitioners had seen this information, as being so static that the dynamic planning of these items via roadmaps had not worked properly. It was not clear for practitioners what would be the right level for describing the solution as a whole. In addition, there were not clear roles or groups of people for managing and planning the solutions as a whole in the case organizations at the time of the study.

4.2.3 Prioritization practices, models and methods

According to our findings, requirements prioritization practices are informal and dependent on individuals in practice. Individual practitioners prioritized requirements on the basis of their experience and tacit knowledge. In the case organizations, the practitioners typically had an experience-based outlook about the most important requirements that should be implemented first. Instead of a prioritization method, they were longing for help in validating their draft decisions. They would have wanted to make sure that they were making their prioritization decisions rationally enough and based on the right aspects affecting the importance of the requirements. However, practitioners used different practices to systematize the informal prioritization. Examples of such practices are described below.

Mutual cost-value analysis. The most commonly used mental model was to sketch the implementation costs and potential value of different requirements. Many of the case organizations tried to estimate a requirement's value for their company and its implementation costs in different phases of the development work. This kind of estimation was performed in order to find the requirements with the best cost-value ratio. The actual estimation was carried out more on the basis of experience and in a mutual fashion than by using strict scales and a defined basis for the estimations. 'We try to judge a requirement's value and costs in the early phases of development. We have no formal method for that,' explained one interviewee.

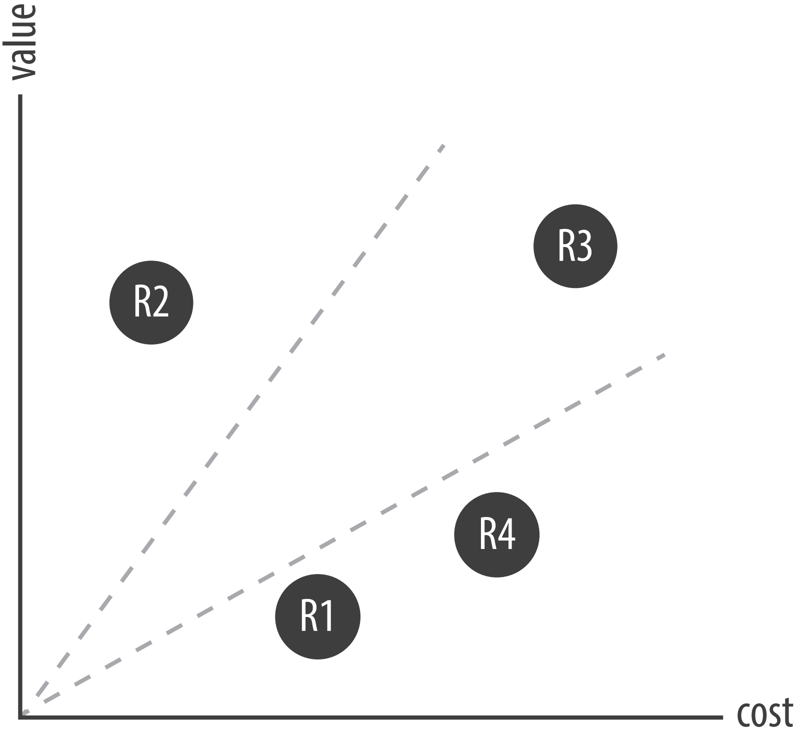


Figure 5. Example of a mental model: Cost-value –analysis

Modified Kano model-based analysis. One of the case organizations used as a mental basis for their requirements prioritization decisions a model modified from the well-known "Kano Model of Customer Satisfaction" (Kano et al. 1984). Kano's idea is basically that customer requirements can be divided into basic, performance, and delighter requirements. Basic requirements are the ones that customers expect the product to have, and are dissatisfied without. Delighter requirements are the ones that customers do not expect, but that lead to delight and considerable satisfaction if they are implemented. The third group, the performance requirements, are requirements that improve something that already exists but that do not have that much of an effect on satisfaction. ‘We try to focus on the requirements that delight most or that have the best effect on the biggest issues on the dissatisfaction side. We try to avoid focusing too much on performance requirements that have less effect on satisfaction, but may, however, require a lot of work to implement.’

Evaluating aspects affecting priorities. Many of the interviewees mentioned aspects affecting priorities that they try to keep in mind and use as a basis when prioritizing their requirements. The aspects affecting the priorities of the requirements, however, seemed to be very case- and company-specific. Examples of such aspects are importance to certain customers, implementation costs, or the requirement’s fit with other products in the product family. It seems that, for example, the size of the company and type of the product and the markets have an effect on the aspects that are important for practitioners to consider in prioritizing their requirements. In addition, projects within companies are different and therefore have different goals and success criteria.

On a high level it seemed that practitioners were usually not able to take into account all of the aspects that they would have seen as being an ideal combination. For example, it seemed that the market situation forced many product managers to focus on pleasing their potential new customers more than their existing customers in order to conclude their sales cases.

Other practices. The practitioners also analysed the impact of different requirements according to certain criteria and they collected additional information from markets or from internal stakeholders to be better able to make the decisions based on real information. Some companies also systemized their requirements engineering processes in order to better involve the right stakeholders in decision-making.

4.2.4 Aspects affecting priorities

Our results indicate that the aspects affecting the priorities of requirements in practice can be roughly divided into three high-level viewpoints. These viewpoints (introduced in Figure 6) that encapsulate the other aspects are ‘Customers and users’, ‘Company’s own business’ and ‘Implementation’. The individual aspects affecting the priorities of the requirements, however, seem to be very case and company specific. Examples of such aspects in the case organizations include such as ‘importance to certain customers’, ‘implementation costs’ and ‘the requirement’s fit to other products of the product family’.

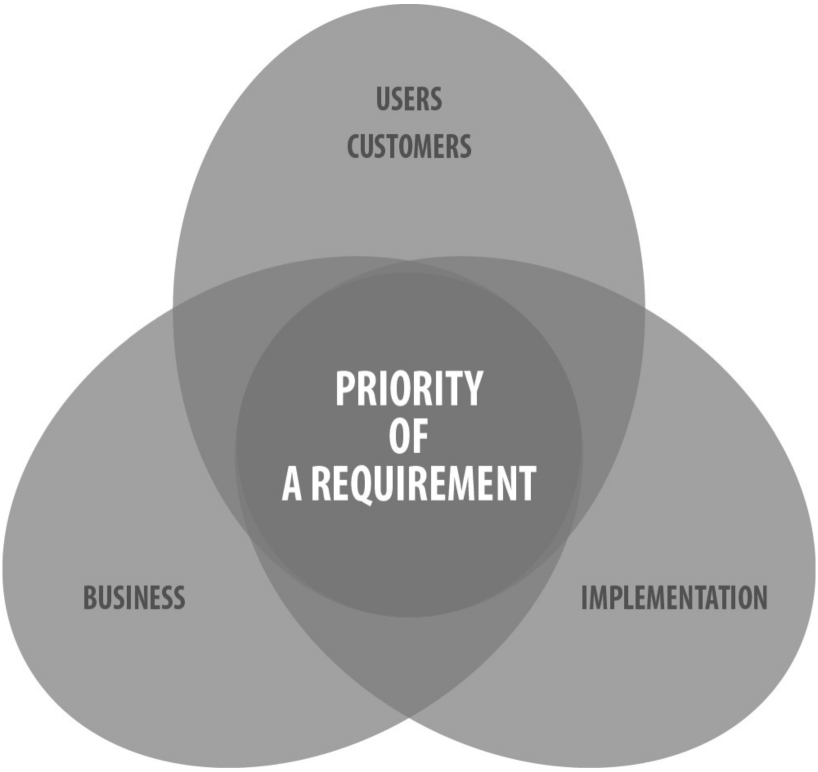


Figure 6. Three main aspects that gather the other aspects affecting priorities: business, customers & users, and implementation

Practitioners tried to keep in mind the aspects that they thought to affect to priorities and used them as a basis when prioritizing their requirements. However, the individual aspects that the interviewees mentioned varied greatly. Our findings also indicate that that the company size, type of the product and the markets have an effect on the aspects that are important for practitioners to consider in prioritizing their requirements. Also the projects within the companies are different and have different goals and success criteria; thus different aspects should be into account in requirements prioritization.

The amount of aspects affecting the requirements priorities seemed to expand in market-driven companies compared to the companies operating in project business. For example, many interviewees in the product management mentioned the satisfaction of intermediaries (such as local resellers or consultants) as one important aspect affecting their priorities, which was not present in project business. A more detailed view to aspects affecting to the priority of a requirement, according to the study presented in Publication I is introduced in Figure 7.

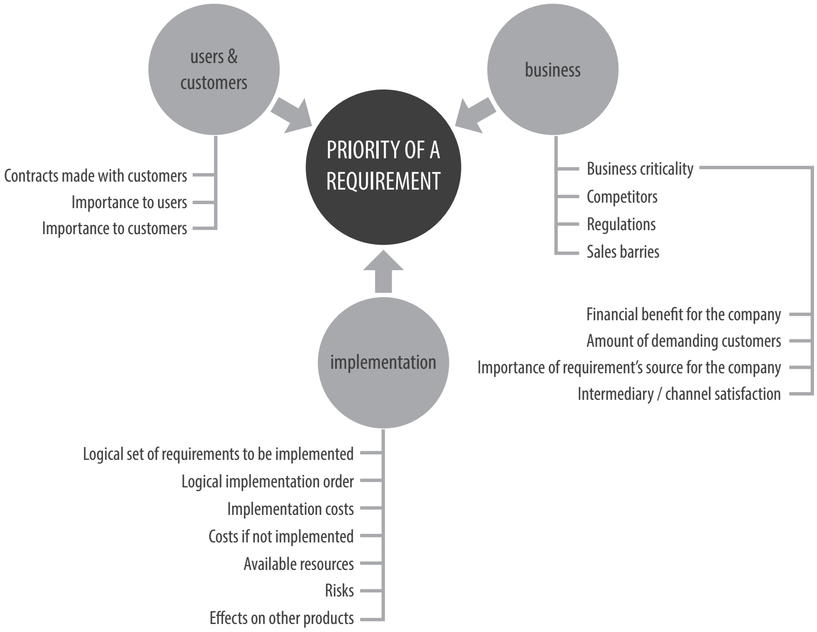

users & customers
business
PRIORITY OF A REQUIREMENT
implementation
Contracts made with customers
Importance to users
Importance to customers
Business criticality
Competitors
Regulations
Sales barries
Financial benefit for the company
Amount of demanding customers
Importance of requirement's source for the company
Intermediary / channel satisfaction
Logical set of requirements to be implemented
Logical implementation order
Implementation costs
Costs if not implemented
Available resources
Risks
Effects on other products

Figure 7. Aspects affecting the priority of a requirement

Our findings indicate that it is difficult to get all of the important information about aspects that influence the priorities of requirements and explicitly draw different points of views together. At a high level, it seemed that the practitioners were usually not able to take into account all of the aspects that they would have seen as being an ideal combination. In the product level, for example, the market situation forced many product managers to focus on pleasing their potential new customers over their existing customers to get the sales cases closed. Furthermore, the practitioners pointed out that in many cases situational aspects dominated the decisions. These included the readiness of different requirements, the fit of a certain product developer for implementing the requirement or personal opinions.

4.3 Challenges in long-term product planning and requirements prioritization

The second objective of this thesis was to identify the challenges in long-term product planning and requirements prioritization. This research objective is mainly addressed in Publication I, which focuses on practical challenges in requirements prioritization. Publications III and IV shed light on this problem from the long-term planning perspective. This section summarizes the practical challenges involved.

4.3.1 Long-term product planning challenges

- Gaining and sharing a holistic, long-term view
- Comparing the development needs of different products
- Involving different stakeholders in the planning process
- Linking business decisions to product planning
- Taking technical and developmental issues into account
- Tying product development resources to roadmaps

Gaining and sharing a holistic long-term view
Understanding the long-term view of products and describing it clearly seemed to be difficult in practice. The case organizations had found that requirements documents for short projects were not sufficient to ensure the necessary kind of wide understanding in the organization. Both the internal and external communication of the products' future development steps and their rationales were critical issues that the companies had experienced problems with.

Practitioners from the case organizations complained that they had problems in seeing the 'big picture', as product planning was typically focused on features of individual products in the relatively short term. By 'big picture' they meant that it was difficult for them to see what the goals of further releases were, and how well the decisions made concerning features matched the needs of different customer segments. Practitioners also complained that roadmaps immediately became out of date.

Furthermore, finding a reasonable abstraction level and amount of information that one roadmap should contain was experienced as difficult. Practitioners in our study found that by writing contents that were too detailed and technical, they lost sight of the business view, while if the information was too rough and high level, it had no use in further development work.

Comparing the development needs of different products
The management in the case organizations faced the challenge of comparing different projects and feature ideas with each other. Typically the product managers were fighting for the same product development resources. This meant that the practitioners needed better ways to communicate their future ideas and resource needs to other product managers in a way that was understandable for others. This communication was also needed to find and realize potential synergies that might exist between different products and their future development directions.

Involving different stakeholders in the planning process
Companies wanted to inform and involve more stakeholders (R&D, marketing and sales, and management) within the company earlier in the development cycle than their existing situation was. In addition, they saw a need to start the development by setting the high level targets co-operatively so that the marketing and sales functions were able to prepare their activities at the same time as the product development activities. However, in practice, the discussions were held in the feature level; therefore, they were difficult to understand for

stakeholders other than product managers or product developers. In addition, the existing processes were not supporting the involvement of other stakeholders than product management.

Linking business decisions to product planning

The relationships between business goals and decisions regarding features seemed to be quite difficult to identify in practice. All of the case organizations felt that the feature-level roadmapping was not enough and that a more business-oriented view of the future was needed. Furthermore, explicit linking between different planning levels was needed to understand which business targets are affected if certain features are not implemented or which products are affected by different business targets.

Taking technical and developmental issues into account

Our findings indicate that in the long-term planning phase the product development viewpoint was less emphasized compared that of other stakeholders. The focus of the long-term product planning was mostly on other rationales and information than that of product developers. However, there were many technical limitations and technical knowledge in the product development that was important from the viewpoint of long-term planning. The challenge in practice seemed to be how to combine the technical viewpoint into product plans that were at the time made without deep technical knowledge.

Tying product development resources to roadmaps

Tying product development resources to roadmaps was experienced as difficult in practice. This meant that practitioners found making realistic plans as to be challenging, as they did not have ways to understand the linkages between what can be implemented with current resources and what the goals are. The resource pool is always limited; product developers have different skills, which have to be taken into account already in the planning phase. Practitioners may feel frustrated if they do not see any chance of implementing all allocated requirements within a given time. Conflicting roadmaps from the resource point of view do not create trust.

4.3.2 Requirements prioritization challenges

- Requirements prioritization is an ambiguous concept
- Manifold aspects have an effect on priorities
- Aspects affecting priorities are difficult to combine
- Every single requirement cannot be analysed in detail
- Describing and communicating priorities and their rationales
- Developers do not know enough about customer preferences

Requirements prioritization is an ambiguous concept

The terms 'requirements prioritization' and 'priority"' had several different meanings in practice without practitioners being aware of that. This caused confusion and misunderstandings amongst product development personnel.

In case organizations, the term ‘requirements prioritization’ had many meanings. Occasionally, the term was used with the meaning ‘Deciding which requirements are the most important ones in the long run?’; sometimes it meant ‘Which requirements we have to implement right away?’ or ‘Selecting the requirements that will be implemented first in this project?’, or ‘Which of the requirements describe the system in high-level terms?’. There were ambiguities in the usage of the term ‘priority’, as well. In some cases the term was used as a quantity meaning ‘the importance of a requirement to the customer’ and in other cases it described how soon the requirement would be implemented.

Manifold aspects have an effect on priorities

Our findings indicate that the priority of a requirement is based on many aspects (called ‘factors’ in Publication I). Amount of aspects affecting the requirements priorities seem to expand in market-driven companies. In practice, there was no time and there were no possibilities to figure out all of the relevant information as a basis for priority decisions.

Aspects affecting priorities are difficult to combine

Our study also indicates that the challenge is not only in selecting which aspects to take into account in requirements prioritization and getting enough information about those. Companies also did not know how to systematically combine different market and customer preferences in practice. ‘There are requirements from this customer and that customer. There are Japanese requirements and requirements from the U.S.A’, explained one product manager on how it its difficult to combine this information. ’Now we are in a situation where the customers who complain most get most’, continued another product manager

Every single requirement cannot be analysed in detail

One big challenge that practitioners experienced in prioritization was that there were not enough resources to analyze every requirement in detail. The development personnel had for example difficulties in analyzing all of the raw requirements they gathered from customers. Requirements management systems were overloaded by requirements from different markets and local offices, but there were no efficient practices to analyse these.

Describing and communicating priorities and their rationales

According to our findings, one important issue in requirements prioritization seems to be the communication of the priority rationales through the organization. It is not only important to know the priority order or getting a priority list from a certain area office; it is also vital to communicate this information and the rationales behind it in a clear matter to different stakeholders.

In the case organizations, the product managers needed to know why the local offices considered some requirements more important than the others in order to make their decisions concerning the future development steps of the products based on different views. In addition, project managers wanted to see the big picture about the priorities and business rationales behind, when planning the milestones of the projects. In general, practitioners felt that there was not enough communication about this background information.

Developers do not know enough about customer preferences
Usually product development personnel had no idea of why certain requirements are important to users because people were working in an isolated manner; product development personnel did not have direct contacts with users and customers. In addition, there were no common practices to communicate customer and user information through the product development process. Particularly in small projects, contact with customers and users was too narrow.

A great deal of important information was gathered from users by the help desk calls they made. Furthermore, product development staff communicated with vendors and gathered information in this manner. In the case organizations, product managers created the first requirements specification on the grounds of discussions they had had with customers. They had an idea of which requirements were important to customers and placed requirements into priority categories. The typical case was that the customers who complained most, got the most.

There were no generally agreed-upon ways to transfer priority information to the project group, and usually, the original reason for requirements being considered important failed to reach as far as to the project manager and the other project group. One participant complained that 'Usually there is no clear explanation besides requirement or need, why it is important or wanted. A person who does not know anything about this particular requirement from the customer point of view does the prioritization.'

4.4 Suitability of requirements prioritization methods

The third objective of the thesis was to analyze how requirements prioritization methods introduced in the literature suit requirements prioritization in practice. This research question is addressed in Publication II where two requirements prioritization methods were evaluated in two case organizations.

The results concerning the evaluation of prioritization methods are introduced in this section and summarized below:

- Methods can structure the discussion and help to take different viewpoints into account
- Methods do not take the limitations of a real environment into account
- The priority list calculated by a method cannot be used as such
- Methods falsely encapsulate the assumption that the value of a release is the sum of its ingredients

4.4.1 Methods can structure the discussion and help to take different viewpoints into account

There seems to be some benefits in using a requirements prioritization method in the product development according to our study.

Firstly, requirements prioritization methods can provide a more controlled and systematic way to evaluate the requirements. In our study, the practitioners found the idea of evaluating the requirements in a controlled way from different viewpoints to be interesting. Especially in the Wiegers' case, where all of the requirements were evaluated according to their value for customers, implementation costs and risks, practitioners felt that they got new ideas about the aspects that should be taken into account in prioritization.

Secondly, the idea of having requirements distributed in descending order as a result was recognized as a benefit. The product developers in both cases were interested in evaluating the prioritization methods because they were enthusiastic about the idea of getting their requirements distributed in a descended continuum as an outcome. However, combining the different stakeholder preferences was concidered difficult from the priority lists.

Thirdly, using requirements prioritization methods may cause positive side effects. In the pair-wise comparison case, the selected users were interested in sharing their preferences concerning the product with the product developers, and they took the practitioners' interest in their opinions as an honor. In addition to the numerical feedback they gave for different requirements by using the methods, they also talked freely about their preferences and opinions about the products during the prioritization work.

4.4.2 Methods do not take the limitations of a real software product development into account

One basic problem with the requirements prioritization methods seems to be that they do not take the limitations of a real software product development into account. Table 16 sheds light on product development in practice versus characteristics and the underlying assumptions of prioritization methods according to the literature study.

Table 16. Product development in practice versus characteristics of requirements prioritization methods

Product development in practice	Characteristic of prioritization methods
Large amount of requirements	Scalability of methods is low
Requirements are not a strictly defined, structured set of items at the same level of abstraction.	Methods require a set of requirements that are comparable with each other
Many aspects affect the priorities	Methods take only couple of aspects into account
Many products are developed at the same time	Methods do not have support for comparing requirements of different products
Development resources are common for different products and different developers can do different tasks	Methods do not take development resourcing into account
Value for the customer is created when he or she uses the product as a whole. Practitioners need to know which set of requirements to publish at a time.	Methods handle requirements as individuals

The scalability of the requirements prioritization methods seems to be quite limited. In our cases it was found that the scalability of e.g. the pair-wise comparison technique in terms of the amount of the requirements was found to be quite low. Pair-wise comparisons with over 20 requirements were difficult in practice. In practice, the requirements management systems in the software companies were overloaded by requirements from different markets and local offices. The ability of requirements prioritization methods to scale in these situations seemed to be quite low.

Requirements of different levels of abstraction cause trouble with requirements prioritization methods. According to our study, it seems that, in practice, requirements prioritization is not comparing a set of structured requirements in the same level of abstraction with each other. The pool of requirements is rarely so strictly defined and clear. The user requirements in our pair-wise comparison case were at quite different levels of abstraction from each other. It seemed that some of the requirements were getting low priorities just because they were not compared to the other requirements of the same level of abstraction. For example, in one requirements group, a more general 'quality' was prioritized over a specific material alternative when they were compared. However, on that basis, it would not be right to indicate that this material alternative could not lead to quality or be an excellent material alternative.

However, the amount of aspects affecting the requirements priorities seem to expand in market-driven companies compared to the companies operating in project business. For example, many of the interviewees mentioned the satisfaction of intermediaries (such as local resellers or consultants) as one important aspect affecting their priorities, which is not that important in project business. Our findings indicate that the evaluated requirements prioritization methods have a limited ability to support decision making in such a complex area as requirements prioritization in market-driven product development.

There is usually a fixed number of working hours that can be used for product development in one release. In many companies, these resources have to be divided for the development of many products. It seems that requirements prioritization methods introduced in the RE literature are not taking these limitations efficiently enough into account. In the methods, requirements are evaluated alone or compared with each other individually. This leads to lists or diagrams of individual requirements. However, the question that typically arose in the case organizations was as follows: 'Is it cleverer to implement this one big requirement or these eight small requirements that do not get very high scores alone?'

4.4.3 The priority list calculated by a method cannot be used as such

Requirements prioritization methods give a priority list as an outcome. These lists may lead to a false impression amongst the practitioners as to what one should do on their basis. In our case, the practitioners found that it was not possible to just select the first requirements from the priority list and be sure that these are the most important requirements that should be implemented first. For example, in the Wiegers' method case, we also found that taking value per cost plus risk ratio may lead to a priority list that favors unremarkable requirements with both low value and costs and undervalues important but expensive requirements (e.g., 2/1 is a bigger number than 8/7).

Furthermore, issues that are unclear in the usage of the prioritization methods may lead to wrong calculations and thereby be incorporated into prioritization results. In our cases, some practical problems occurred during the prioritization work. We found that these affected the priority order of the requirements. Problems that occurred during the cases were such that users found it difficult to estimate how much more valuable one requirement is than another; it was unclear for practitioners on what information they should base the evaluations of the aspects; and practitioners found it difficult to estimate which number to give to aspects.

Getting exact numbers or fractals as an outcome does not ensure the validity of the results because the mathematical calculations cannot improve the quality of the raw data inserted. If a user evaluated the aspect in a hurry without careful consideration, or could not nominate the extent to which one requirement is more important than the other or if a practitioner did not know what a requirement's real value for the customer was, the prioritization results were nothing more than just rough guesses.

Furthermore, practitioners seem to mistrust the results they get by using prioritization methods. In our cases, the practitioners were interested in using the methods and felt that they would need one to make better prioritization decisions. However, if the priority order given by the method was contrary to their experience-based opinions, they felt that the method was not working properly which made them manipulate the original raw values to get the 'right' outcome.

4.4.4 Methods falsely encapsulate the assumption that the value of a release is the sum of its ingredients

The implicit goal setting in the requirements prioritization methods introduced in the literature is to 'put requirements in order according to their importance'. In requirements prioritization methods, the importance of each requirement typically comes from the separate analysis of each requirement. The basic idea in the requirements prioritization methods seems to be that all of the requirements are processed uniquely according to their value after which the most valuable requirements are then implemented. This means that from the viewpoint of a prioritization method, each requirement typically is its own unit with some kind of a value associated to it. This is implied to lead to the best added value with the product release.

However, according to our study, value is not in each requirement as such but is created in the process of the user using the system and thus from the interplay of the requirements. This means that by selecting the individually most valuable requirements to be implemented may not lead to maximizing the value of the product.

4.5 Supporting the link to customer value creation

The fourth objective of this thesis was to analyse how to support the link from long-term product planning and requirements prioritization to customer value creation. This research objective is mainly addressed in Publication IV, which introduces practices that help to link requirements prioritization to customer value creation and in Publication V, which analyses the role of RE practicesin customer value creation. Also Publication III sheds light to the issues, as long-term product planning was found out to link business view into the product development.

4.5.1 Explicating planning levels, horizons and the rhythm of planning

Our study revealed a jungle of different planning documents and decision-making forums in the case organizations. There were for example business plans, long-term product plans, technology roadmaps and requirements documents for individual projects. There was no clear picture of what was actually planned, how long onwards and how often. This meant that practitioners did not have clear picture of which documents, by whom and how often different aspects of products' future and different aspects of customer value creation was discussed within the organization.

Our results indicate that companies need to analyse and explicate the planning levels and time horizons that they need for planning their products' future as well as how often different plans need to be discussed and updated. The term 'planning level' refers to items that are planned, while the term 'time horizon' refers to the length of the time period for the plans. The latter should be made explicit for each planning level. An example of the analysis of planning levels and time horizons in one case organization is presented in Figure 8.

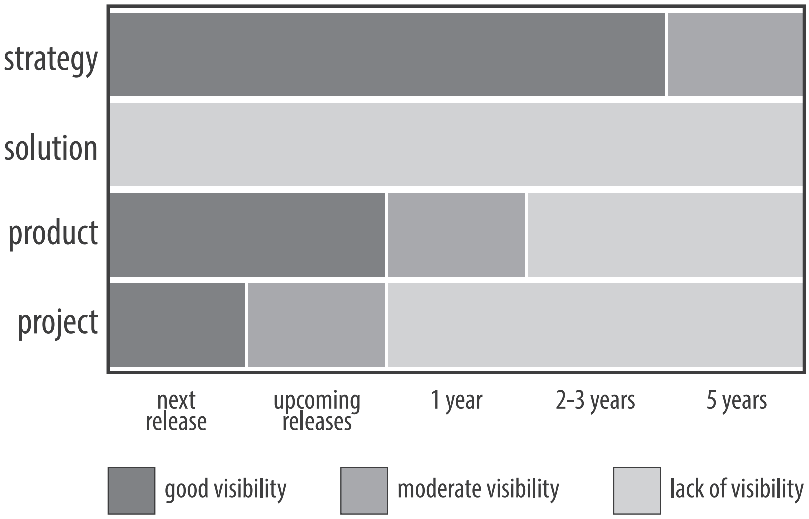


Figure 8. An example of the planning levels and time horizons in one case organization

In practice, this means that companies should discuss questions such as the following: 'What, in our case, needs to be planned and for how long forward?', 'What roles should participate in the planning and how often?' and 'What kind of documents we produce?'. The relationships between different planning levels and the corresponding responsibilities should be visualized to help practitioners to understand them and the gaps in the current practice.

Our findings indicate that it is preferable to plan in an open-ended manner but with a predefined rhythm (see Figure 9).

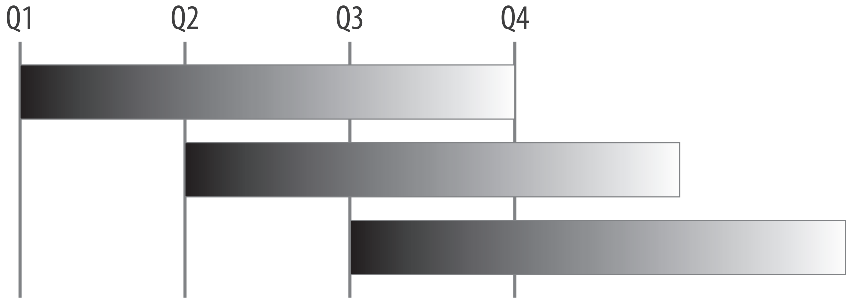


Figure 9. Planning open-endedly with a predefined rythm

4.5.2 Adding intermediate planning levels between business and requirements engineering

Our results indicate that adding intermediate planning levels between business decisions and requirements engineering helps to link them together and to sustain the customer value viewpoint during product development. For example, marketing arguments are easier to tie to high-level features than to individual, small-scale requirements. In addition, gaining and sharing a holistic long-term view of the future with different stakeholders is important for being able to combine the three viewpoints (the company's own business, customer and users, and implementation) and for the co-ordination of resources over time. For example, sales and marketing need early information about future developments and the technical viewpoint is easier to take into account, if architects and product developers can take part in the discussion early enough and already at a high level. Furthermore, explicit links between the development needs of different products are needed, as product development resources are usually shared.

4.5.3 Discussing product's business goals and resource allocation separately

Our findings indicate that business goals for the products should be discussed separately from R&D resource allocation. According to our findings, it seems that at least two distinct levels in long-term planning are needed to both be able to both see the big picture from the business and customer viewpoints, and manage the product development resource allocation.

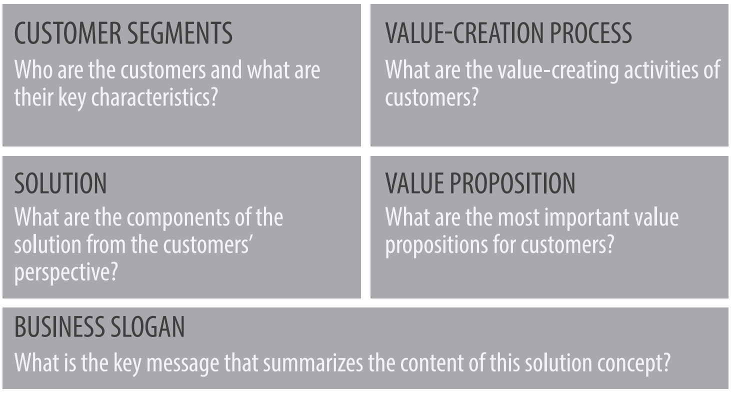
Market-oriented or commercial roadmaps form an overall view of the offering and give an understanding of how well different customer segments would be served in the future. On this level, the roadmap(s) might depict issues such as the needs of target customer segments, the positioning of the products, and the different market trends for the next few years. These roadmaps serve as a basis for more detailed planning. On the other hand, lower-level release or software roadmaps are needed from the perspective of managing software development. These roadmaps are needed for allocating product development resources to different software products and for giving product developers additional information.

4.5.4 Making solution planning visible

The interviews with the stakeholders and the gap analysis of the existing planning levels and horizons in the case organizations revealed that there was especially a need to gain and share a holistic long-term view of the solutions. Usually, the view of solutions that were provided to customers seemed to be a blind spot for the companies, as the planning was primarily just the product managers' task that focused on software components and features. In the context of the company, a solution means a comprehensive set of software and service components that are required to fulfil customer needs.

On the basis of the interview results, the researchers proposed solution concepts for the case organization. In addition to supporting different stakeholders in solution planning, the purpose of concept descriptions is to provide a big picture of the solution for all employees and guide requirements engineering activities so that the customer viewpoint gets explicated.

The contents of solution concepts were processed and evaluated with different stakeholders in the workshops. Figure 10 illustrates the five elements of the solution concept. The key principle of the solution concept is that it should be short and visual. The five elements of the solution concept (Customer segments, Value creating process, Solution, Value propositions and Business slogan) are introduced in more detail in Publication V.



CUSTOMER SEGMENTS
Who are the customers and what are their key characteristics?

VALUE-CREATION PROCESS
What are the value-creating activities of customers?

SOLUTION
What are the components of the solution from the customers' perspective?

VALUE PROPOSITION
What are the most important value propositions for customers?

BUSINESS SLOGAN
What is the key message that summarizes the content of this solution concept?

Figure 10. The five elements of the solution concept

4.5.5 Emphazising solution thinking

Our findings show in general the importance of widening the planning scope from the features of individual products to solution planning in order to support the customer value creation. We use the term ‘whole-product thinking’ to refer to this kind of widened planning scope or, more generally, to address different stakeholder concerns in different activities. In this section, we summarize the practices that help the organizations to internally support solution.

Identifying customer segments

The practitioners in the case organizations saw that customer segmentation is useful, especially when the number of customers is in the hundreds or thousands. The basic idea of customer segmentation is to analyze the existing and potential customers of the solution, identify homogenous groups and describe the main characteristics of each group. Based on the analysis, the solution can be customized for each customer segment. Customer segmentation can, however, be difficult in practice. For example, finding reasonable factors to be used as a basis for the segmentation and understanding what is a reasonable size of a segment is not trivial according to the practitioners. Segmentation is a common practice amongst marketing personnel but is often not integrated with RE activities.

Forming the customer segments is not enough. The results of the segmentation must also be used when creating the solutions for customers. Ideally, both the core product and all of the different service components should be customized to support the customer’s process. In practice, many companies rely on features to realize differentiation amongst segments. This may lead to a product portfolio where products are different but not in a way that is meaningful to customers.

Discovering information about customer processes actively

According to our findings, the companies see customer information collection as important. Typical ways to gather feedback are customer satisfaction surveys and discussions with the customers. The companies also receive error reports and other passive feedback from customers.

All of the companies have recognized the need to emphasize the collection and analysis of customer information. Some of them have established groups that focus on customer information gathering and sharing. In some companies, the usability group has taken an active role in doing user studies and spreading this information to other organizational units. Some of the companies have also created automatic ways for collecting and analyzing how users use the product via tools that record the actions that users make.

In some companies, observing how customers behave in their natural environments has provided deeper insight into the customers’ processes than asking for their opinions in isolation. Nevertheless, observation and ethnography are still seldom used in the companies.

Creating direct contacts between developments engineers and users

Traditionally product developers have had difficulties in understanding customers' processes thoroughly bacause they have received the information via other people, such as product managers, and because they hardly ever met customers and users.

Some of the case organizations have actively created direct contacts between development engineers and customers and users. Development engineers have had for example possibilities to familiarize themselves with the customers and their working facilities. Developers have also interviewed customers about their needs and collected feedback directly from the end users via prototypes. This has provided them a better understanding of users and their processes.

Establishing a cross-functional planning team

Our findings indicate that planning and launching a software product release needs cross-functional effort and outputs from functions other than R&D as well. For example, marketing material, product documentation, and sales campaigns need to be planned and implemented before a product can be launched as a whole product. This requires cross-functional effort even in the early planning phase.

During the study, the two main case organizations concluded that they needed a cross-functional project team for planning the main product release launches from the early phases of development. The project groups consisted of members from marketing, product management, documentation, and service. The group gathered once a week throughout the product development project. During this time they made preparations for launching the release.

The practice improved their ability to synchronize other functions with R&D and also to prepare outputs other than software earlier than before. The other case organization even changed its organizational structure from a functional one to a more market segment-based one. In this new structure, every market segment had its own segment team that included representatives from many internal functions (product management, sales, marketing and service). The team was jointly responsible for integrating segment development plans into the overall product development plan. The practitioners felt that this type of organization brought experts from different functions nearer to each other and improved their communication. In addition, the shared responsibility forced the practitioners themselves to truly co-operate with each other.

5. Discussion

This section presents the main findings of the study and compares them with the results of previous research. The findings are discussed here according to the research questions, which were formulated as follows:

1. What is the state of the practice in long-term product planning and requirements prioritization?
2. What are the practical challenges in long-term product planning and requirements prioritization?
3. How do requirements prioritization methods introduced in RE literature suit for solving practical challenges?
4. Which practices support linking long-term product planning and requirements prioritization to customer value creation?

At the end of the section, the limitations of the study are discussed.

5.1 State of the practice

Our study indicates that, in the software product development context decisions regarding what features will be implemented in the products are made in iteratively in many phases of product development. In the literature, requirements prioritization is traditionally described as a part of the requirements analysis phase (Sommerville 1996). Our study, however, indicates that instead of being just a one-off activity, requirements prioritization is needed in many phases of the development work. Recently, other authors have also reported the positive side effects to value creation from prioritization not being just one off activity (Racheva et al. 2010b).

According to our study, there are two main levels in software product development, in which prioritization decisions are mainly made. On the *product level,* the goal of the prioritization is to determine the features to be implemented in the product(s) in the long run so that the maximum amount of value is provided to customers and users and thus to the company itself. On the *project level*, the main goal of the prioritization is to organize the requirements that are already selected for a project into a sufficiently rational implementation order. At this phase, technical issues and for example the maturity of requirements affect the priority order of requirements. The importance of product-level prioritization is noted also by Ebert (2005), who reports that in order to provide value with their products, companies need to place an emphasis on the selection and prioritization of requirements also before projects in addition to within-project activities.

According to our findings, the priority of a single requirement is based on many aspects. These can be such as the financial benefit of the requirement for the company, the requirement's importance to users, and implementation costs. Our findings suggest that the aspects affecting priorities can be grouped under three main points of view: business, customers & users, and implementation. Other authors also discuss the importance of combining different aspects in requirements prioritization, but our study provides deep analysis about main viewpoints that capture the aspects and their roles in different phases of product development. These findings partly support Khurum et al. (2013) who present a consolidated view (called the software value map) of the software value concept utilizing four major perspectives: the financial, the customer, the internal business process, and the innovation and learning. Their value map is supposed to offer a unified view of value where value concepts are categorized as value aspects, sub-aspects and value components, which can be used by professionals to develop a common understanding of value, as well as acting as decision support to assure no value perspective is unintentionally overlooked when taking product management and development decisions (Khurum et al. 2013).

Our study indicates that the number of aspects that affect the requirements priorities seem to expand in market-driven companies compared to the those companies that operate in project business. For example, the satisfaction of intermediaries (such as local resellers or consultants) is one important influencing aspect, which is not present in project business. Furthermore, many aspects that affect priorities seem to be situational and the importance of different aspects seems to also depend on the development phase. Our study indicates that in the early phases of development, business issues have more influence on priorities and as projects evolve the technical and resourcing constraints play a bigger role when deciding the implementation order.

Our findings indicate that on the *product level* long-term product planning seems to be mostly product managers' individual responsibility. Both product-level and project-level activities seem to require expertise from many stakeholders that should be able to combine the user and customer needs and the company's long-term business goals with the product development constraints. Furthermore, our findings indicate that practitioners also experience a need to involve other stakeholders, such as sales, marketing, and management more tightly and earlier in the long-term product planning. Also, for example, Moisiadis (2002) argues that prioritizing requirements should involve representatives from each group of stakeholders with a vested interest in the success of the development project. However, in practice, involving different stakeholders and viewpoints in different levels in product development is not systematic.

Our findings show that focus of long-term product planning in market-driven product development is on the features of software products. Both planning practices and design documents typically **focus on features of one software product at a time**. For example, roadmaps and release plans are made only for the core software and the linkages between different products are not explicit even on the long-term product planning level. Furthermore, our results indicate that the solutions for customers are not planned as entities in software product development companies. In practice, this means that the solutions do not have owners and the planning processes do not support the planning or discussion of solutions as a whole. The other perspectives of value creation, such as customers' processes that the product shall support in the future or the services related to product, are not explicitly described in the planning documents or discussed in the planning forums. This makes it more difficult for practitioners to understand the value they are providing for the customers.

The planning horizon in long-term product planning is typically relatively short; i.e., typically one or (at most) a few releases ahead. Furthermore, different level plans are not combined in practice. There are business plans for the business development, product roadmaps for describing the future of different products and technology roadmaps for technical development, but no clear connection between them. Similarly, Groenved (1997) report organizations with a functionality-oriented culture often have difficulty starting roadmapping. They tend to draft only independent, functionally oriented roadmaps (e.g., technology roadmaps).

In general, our study shows that in practice requirements prioritization practices are informal and dependent on individuals. Individual practitioners prioritize requirements on the basis of their experience and tacit knowledge. This finding is somewhat supported by Svensson et al. (2011), who found in their study about quality requirements that ad-hoc prioritization and priority grouping of requirements are the dominant prioritization methods. However, our study indicates that practitioners have mental models that systematize informal prioritization. Examples of such practices and mental models are such as mutual cost value analysis or requiring local offices to write rationales for the priorities they provide.

Our findings also indicate that practitioners make decisions about priorities without explicitly being aware which aspects they take into account and to what extent. This finding is supported by Svensson et al. (2011) who report that it is common to use e.g. customer input as criteria for prioritization but that the absence of any criteria is also common. According to our findings, however, practitioners have attempted to keep in mind the aspects that they believe to most affect priorities and use them as a basis when prioritizing their requirements.

One of the key results in the area of current practices is our finding that, in practice, prioritizing requirements or features is not by nature comparing a set of requirements in the same level of abstraction with each other. This is somewhat contradictory to the research in the area of requirements prioritization where 'work on release planning (and requirements prioritization) has focused on developing model-based approaches designed for a situation where there is a single product/service offering with a set of possible features to be selected from' (Svahnberg et al. 2010). Our results show that the pool of requirements is not so strictly defined and clear.

Our findings also indicate that communication of the priorities and priority rationales through the organization is a key issue in requirements prioritization. Product managers need to know why for example resellers or sales personnel in the area offices of the company consider some requirements more important than the others in order to make their decisions concerning the future development steps of the products based on different views.

5.2 Practical challenges

5.2.1 Long-term product planning

Our findings indicate that long-term product planning in the software product development context involves manifold challenges. These challenges are mainly related to the difficulties of combining different viewpoints, inability to form and share a real long-term view with different stakeholders, and the focus of the planning being too narrowly set upon the features of the products.

Our results show that companies have difficulties in drawing different viewpoints together during the long-term product planning phase. The difficulties of combining different viewpoints in the long-term planning have been discussed by other authors as well (Jantunen et al. 2006). According to our study, one of the main challenges seems to be that the viewpoints stakeholders, other than product managers, are left with too little emphasis. Our study shows that long-term product planning is too strongly product manager-focussed in software product development companies and that current processes do not support involving different stakeholders in planning.

In particular, combining the business knowledge of product managers with the technical knowledge of project managers and architects, in different phases of planning seems to be difficult in practice. One of the well-known challenges in the literature is that taking business decisions into account in product planning is not easy. This challenge was also present in our study, since practitioners do not know what different business decisions should mean for the product's future in practice.

Our study revealed that in addition to the challenge of integrating business knowledge to the selection of features, another challenge is also how technical limitations and possibilities could be taken into account early enough in the long-term product planning. For instance, Bjarnason et al. (2010) discuss partly the same issue by pointing out that one reason for overscoping in software projects is that requirements have not been agreed upon with the development team in the planning phase and that those people that make the decisions do not know the development capacity.

Our study indicates that the linkages between business and technology when talking about software products are manifold as well as difficult to understand and describe in practice. Therefore business, product, project and technology plans as well as plans regarding different products are not explicitly connected together. This causes a challenge that individuals plan the future, both horizontally (different products) and vertically (business, product, project, technology), in silos. Also Ebert (2009) reports that we too often face silo organizations in marketing, where product management and engineering don´t work together. Karlsson et al. (Karlsson et al. 2002) as well found the balance between marketing and developers´ requirements decisions as a dilemma.

Our study also shows that the focus of long-term product planning seems to be in product features, not in the product as a whole. This makes it difficult for other stakeholders besides product managers (who know the features best) to participate in the planning process. Also (Khurum et al. 2013) found that practitioners in the product management and development have different definitions and understanding of value constructs. Involvement of distinct perspectives does offer an opportunity to look at a decision from different aspects; however, this gives rise to misinterpretations and misunderstandings about value constructs to be considered (Khurum et al. 2013). Our findings also indicate that since the product development items tend to be discussed at such a low level, practitioners other than product managers have difficulty following the discussion and bringing their knowledge about the customer value to the table.

5.2.2 Requirements prioritization

In the area of requirements prioritization, our findings support many of the reported key challenges introduced in the literature. However, our results also shed light upon some new areas in the field of requirements prioritization challenges.

Our findings support the earlier reported findings (e.g. (Berander 2007; Carlshamre 2002)), that taking different aspects into account in requirements prioritization, is challenging. According to our study, it seems that it is difficult, even impossible to gather all the information about aspects that influence the priorities of requirements and explicitly draw different points of view together.

Our findings indicate, that in market-driven product development, many aspects affect the priorities of requirements and practitioners do not know which ones to take into account and to what extent. This supports e.g., Racheva et al. (2010a), who discuss the challenges in balancing between client's and developers' value-creation perspectives. Furthermore, it is not easy, in practice, for practitioners to incorporate the aspects of the decision-making process. For example, business viewpoint is widely agreed to be one of the most important aspects to take into account, but it is not easy for practitioners to know what different business decisions should possibly mean from the individual product feature viewpoint. Also Regnell et al. (1998) reported as one of the key challenges in a market-driven company to relate the continuous prioritization of incoming requirements to a long-term product strategy for a range of market segments.

Our findings support the understanding that the amount of requirements and the dependencies between them cause serious trouble in requirements prioritization (e.g., (Karlsson et al. 2002) (Carlshamre et al. 2001)). According to our study it is not possible to analyse every requirement in detail in practice. Furthermore, our study showed that priority lists are hard to utilize, since there are dependencies between requirements and since the customer value of a product is not a sum of the values of individual requirements. Therefore getting requirements 'in order as individuals' do not seem to actually solve the problem of which requirements to implement to provide maximum customer value. All these findings together indicate that seeing requirements prioritization as a paradigm in which individual items are ordered in relation to each other may be a wrong approach.

Our study also revealed some new challenges that are not thoroughly discussed earlier.

Firstly, requirements prioritization seems to be an ambiguous concept. Our findings showed that it is a term that is used for different purposes. On some occasions, it means finding most valuable requirements; on others it means deciding the implementation order. Especially, the shift from product level to project level seems to cause misunderstandings and loss of customer value viewpoint during the product development if the meaning of the term and scales that will be used are not discussed. Furthermore, we found that requirements prioritization really is a different task with different goals in different situations. Therefore, finding a general procedure for requirements prioritization seems to be according to our findings very difficult.

Secondly, our findings underline many of the important background issues surrounding requirements prioritization that have not been largely emphasized as an important part of requirements prioritization to date. Practitioners who make lower-level decisions really need more information about customer preferences and the rationales regarding earlier high-level decisions in the product planning process. It is not enough for product developers that the product manager knows why a requirement is important and assigns the priority with a number or one word that indicates the importance. Also, Regnell et al. (2001) touched on these issues by introducing a visual way to illustrate differences among the importance of different requirements for different stakeholder groups. The focus in the literature, however, usually seems to be on prioritizing a set of requirements, not in how to communicate or visualize the results or how to work on the basis of the results. Furthermore, our results indicate that the communication between the different parties is difficult if it is done only on the level of detailed product features. Therefore our results show that the priorities should be discussed not only on the feature level, but also from the viewpoint of high-level targets for the product.

Thirdly, requirements prioritization has traditionally been discussed in the context of one product or one project. However, in market-driven product development companies the case typically is that there are common product development resources for different products. Practitioners need to make decisions which products to prefer and how to divide the product development resources between the products. The challenge of requirements prioritization in practical cases it is typically not only how to compare the features of one product, but how to handle the whole product portfolio and their priorities.

5.3 Suitability of requirements prioritization methods

Our findings indicate that the evaluated requirements prioritization methods have a limited ability to support decision-making in such a complex area as requirements prioritization in market-driven product development. Having systematic requirements prioritization practices is a challenge because requirements prioritization requires a great deal of non-trivial decision-making.

Requirements prioritization methods can structure the discussion and help take different viewpoints into account and results of the prioritization may work as a basis for discussion. Similarly, Karlsson et al. (1998) discuss positive side-effects that occured during their pair-wise comparison sessions, such as identifying ambiguous requirements. However, the prioritization results should be taken more as being indicative than as a final result that can be used as such. Practitioners seem to also, by themselves, mistrust the results they achieve through the methods. The experiments conducted by Lena Karlsson et al. (2004) revealed similar findings. Some of the evaluators of the requirements, in their case, felt a loss of control over the prioritization process when they used the analytical hierarchy process (AHP). Practitioners in the earlier study also pointed out that models in general may not be useful if it is too time consuming to collect and specify the needed input or if the model is too complicated and complex (Svensson et al. 2010).

According to our study, requirements prioritization methods do not take the limitations of real product development environment into account. Laurent et al. (Laurent et al. 2007) support our findings by noting that 'though the implementation of these techniques is simple, they do not support negotiations and higher level goals strongly'. Methods typically require as an input a limited set of requirements that are not interrelated with each other. Methods also take only a couple of aspects into account, which seems to oversimplify the real product development situation too much. This supports Benestad et al. (2011) who report that models focus on only select parts of a possibly larger space of relevant planning factors. Carlshamre (2002) also reports the limited possibilities that exist of defining, in advance, the aspects that have an effect on priorities. Furthermore, prioritization methods do not have support for comparing requirements of different products with each other or take the development resourcing viewpoint into account in prioritization (Laurent et al. 2007).

Our results indicate that requirements prioritization methods also seem to falsely encapsulate the assumption that the value of a release is a sum of its ingredients. The basic idea in most requirements prioritization methods is that all requirements are processed uniquely according to their value after which the most valuable requirements are then implemented. This should lead the best added value in the product release. However, according to our study, value is not in each requirement as such but is created in the process of the user using the system. This means that selecting the individually most valuable requirements to be implemented may not lead to maximizing the value of the product. For example, Racheva et al. (2010b) report that e.g. the concept of negative value plays a significant role in many projects

Our results also indicate that the challenges in requirements prioritization are mainly other than putting a clearly defined set of requirements in order than assumed in requirements prioritization methods. Practitioners in the software companies seem to have deep experience-based opinions about the priorities of the requirements. Interestingly, we found that instead of exact priority lists, practitioners seem to search practices for validating their tacit knowledge and experience, as well as communicating the priority information through the organization with different stakeholders. This is somewhat contradictory for most requirements prioritization research papers, in which authors try to solve the requirements prioritization problem by developing prioritization methods. However, Svensson et al. (2010) recently noted somewhat similar findings by pointing out that although it may be possible to define release planning as a mathematical optimisation problem, it may not be worthwhile to apply complex mathematics or advanced computational algorithms to achieve ‘optimum’, if the input data to the optimisation process is highly uncertain anyway. Also Bakalova et al. (2011) report based on their mapping study that that there is ‘deviation between the existing methods as prescribed in literature and the processes we observe in real life.’

5.4 Supporting the link to customer value creation

Our results indicate that the creation of customer value with a software product is a complex issue and the selection of ‘right’ features is only a part of it. Also Barney (2008) discusses slightly similar issue by pointing out that since the creation of software product value through requirements selection is not very well understood, it cannot be managed in the most effective way.

Our results, however, indicate that there are practices that strengthen the link from long-term planning and requirements prioritization to customer value creation.

In general, our study indicates that it is beneficial for an organization to *explicate the necessary planning levels, time horizons, and rhythm for planning*. When practitioners analyse and clarify different kinds of plans and planning between strategy and software development, it helps them to understand the potential gaps in their planning processes between business decisions and requirements engineering. Open-ended planning with a predefined rhythm seems to suit for market-driven planning where decisions are more likely ‘now’s and ‘later’s, not ‘no’s. Similar findings were earlier discussed in the context of small companies (e.g. (Rautiainen et al. 2002) (Vähäniitty et al. 2002)).

According to our results, *adding intermediate planning levels between business decisions and requirements engineering* is needed to link them together and to sustain the customer value viewpoint in product development. This supports Karlsson et al. (2002) who argue that co-operation between different stakeholders in market-driven companies requires other ways of communicating than low-level requirements. Our results show that gaining and sharing a holistic long-term view of the future with different stakeholders is important for being able to combine the three main viewpoints (business, customer & user, and implementation) and for coordination of product development resources over time. Already Groenvedl (1997) reported that instead of planning-oriented product development activities, a long-term view is required, based on close cooperation among all disciplines.

Our findings showed that, when *separating the planning of the product business goals from R&D resource allocation and detailed feature-level planning*, it is easier to see the big picture from the business and customer viewpoint, discuss in business terms with different stakeholders, and focus on long-term planning. This also avoids entanglement in the details. Also Ebert (2006) points out that if stakeholders only see a loose assembly of requirements they naturally choose some and start changing. Our findings support Vähäniitty (2005), who also suggests separating different levels of planning from each other.

Emphasizing solution thinking in the organization helps practitioners understand their products and their value to customers more broadly than just as software features and also helps them understand other value sources besides software. It also helps the other stakeholders, such as product developers, marketing and services, to better integrate into product planning that has traditionally been seen as only product managers' responsibility. In this area, our findings indicate that organizing a *cross-functional project team* for planning the main releases of solutions supports the customer value creation by enabling the combining of different viewpoints early enough. Ebert (2005) revealed a similarish finding by reporting that installing a core team for each release affected the success of the product. In addition to this, our results indicate that not only product managers, but also product developers should have *direct contacts with end users to understand their needs, processes and how they use the produc*t.

According to our results, software product development organizations should also *make the solution planning visible.* This means crystalizing the customer value aspects, such as the value creating process, customer segments and components of the solution, and visualizing those with a simple and understandable manner. For example, actively using *customer segmentation* during product planning, not just leaving it as a marketing tool, helps practitioners to sustain the customer value creation viewpoint. If the differentiation of products happens only implicitly based on differences in product features, the value that the product creates for different customer segments is not optimal. Similarly Komssi et al. (2011) propose that companies should shift their focus from the prioritization of software features to the analysis of customers' processes and the prioritization of customers' activities.

5.5 Evaluation of validity threats

Yin (2009) suggests for a case study to evaluate the possible threats to the validity of the observations and conclusions. According to Runeson et al. (2012), validity must be addressed during all phases of a case study. However, it cannot be finally evaluated until the analysis phase (Runeson et al. 2012). In this section, we evaluate the validity threats of our study according to the categorization by Runeson et al. (2012) (summarized in Section 3.9 Validity).

The threats to **construct validity** regarding research question 1 included whether we were able to gain a representative image of the pros and cons of using prioritization methods within the case organization. Regarding research questions 2, 3 and 4, the threats to construct validity included the question of whether we were able to gain a representative and true image from the case organizations' current practices and practical challenges.

As described earlier, to increase the construct validity of the study, we triangulated data sources and data collection techniques. We selected different informants who represented different roles and collected data with different techniques, by different researchers and over a long period of time. However, it might be that we were not able to recognize all the important stakeholders affecting case organizations' long-term product planning and requirements prioritization. For example, company-external stakeholders such as customers and users were not interviewed at all, even though they are important sources for feature ideas and even though the company tries to satisfy their needs with the products. It would be interesting and valuable to gain more information on how the companies incorporated their opinions into the decision-making process. Leaving customer and users without "own voice" in the study left one important viewpoint about the quality of the decisions and the effect of current practices uninvestigated. That posed a validity threat to the study.

In addition, it might be that the interviews, as a primary data collection technique, were limited in their ability to reveal important aspects of the current industrial practice. Selecting requirements to be implemented into a product is a complex decision-making activity, as our study revealed. Practitioners' ability to describe and even verbalize this decision-making process and its challenges might be quite limited. However, designing cases so that they were partly built on each other improved our abilty to delve deeper into those areas in later cases, which seemed important, but fuzzy in earlier cases

Furthermore, interviewing as a technique has always a threat to accuracy of what interviewees say. It may be that the interviewees are not honest or do not properly understand what is asked. However, we think that the risk of dishonesty was reduced in our cases, because the case companies intended to develop their practices, and because we encouraged the practitioners to talk about the challenges freely. Because interviews were semi-structured, not totally structured, we were able to understand the phenomena more deeply, as we were able to ask more questions about the issues that seemed unclear at first. Our extended cooperation with the informants helped us to validate issues that we possibly did not understand earlier.

The threats to **internal validity** include the question whether there were any factors that affected our findings in the case organizations. Our study consisted mainly of interviews. In interview studies, there is always a risk that the researcher biases the results by affecting the interviewees somehow. This risk was reduced in our case by constantly reminding interviewees that there were no right or wrong answers and that we did not expect to hear about perfect practices, but instead wanted to know in detail about what was actually happening so that we could cooperatively develop better practices. We also created a relaxed atmosphere by reminding practitioners that all companies we have seen have struggled with many issues.

We also might have created a certain bias because the companies were operating with each other in the seminars. Furthermore, some of the case companies may have had a similar attitude towards improving their practices in the first place (e.g. that “something is wrong”), which may have been the reason that they became members in our research project. Also the prolonged involvement of the researchers may have caused a similar risk of bias (see (Runeson et al. 2012)).

In general, the researchers’ participation may also have affected validity of the results. However, in all phases of the study, especially in the research design and data collection phases, the main researcher was not working alone. Other researchers validated the interview outlines, contributed to research designs and participated in interviews. In the data analysis phase, however, the author of this thesis performed the analysis alone, which unfortunately left some kind of threat to the study’s validity.

The threats to our study’s **external validity** include whether, on the basis of the situation in the case organizations, it is possible to draw conclusions that concern other software companies operating in the product business. To improve the external validity of the research results, this study involved seven separate case organizations of different sizes. In addition, the companies represented different types of products, customers, and business environments.

Our research was case-based research. Our research was not sample-based research, in which, samples of population elements are studied and generalizations are made about the distribution of variables over the population by means of statistical inference from a sample (Wieringa 2014). Instead, we generilized by analogy to the population of similar artefacts (Wieringa 2014). According to Wieringa (2014) analogic generalization is qualitative, meaning that the researchers indicate to which other cases, or populations of cases, an explanation can be generalized, without indicating to how many of them can generalized. In case-based research, the variability of the real world is reduced by decomposing a case into components, which can produce case phenomena through their interactions (Wieringa et al. 2015). Generalization should be based on architectural similarity, and generalizations can be made more robust by analytical induction (Wieringa et al. 2015). This means that in our research, the validity is not about the sample in relation to the universe, but about the plausible explanations we were able to make from the cases and understand the architecture in which the results are generalizable.

However, the “sample” of the studied companies still has some kind effect on the plausibility of the study. All the case organizations were Finnish companies, and therefore there might be issues that may not be appropriate to other cultures. In addition, our co-operative relationship with the case organizations creates the potential for a bias. The case organizations were selected using a convenience sampling strategy. The organizations were industrial partners in our research projects, which implies that they considered long-term planning, requirements prioritization and RE in general areas that are both essential and need improvement. However, we believe that the case organizations are typical market-driven software development companies.

Especially, regarding research question 1, threats to external validity are mostly because we had only one case organization, only two case projects, and only 20 requirements to prioritize in both case projects. However, even with such a small sample of people, requirements, and methods, we were able to find common denominators from both of the cases (by varying these different variables) that impeded the practitioners’ willingness to use the evaluated methods. It was still possible to provide plausible and thus generalizable explanations for the challenges and benefits by using the prioritization methods in that kind of setting were such as they were.

Some other authors have evaluated methods in the laboratory or with university students as subjects (Berander 2004). There is value in conducting such studies; they may lead to more exact results concerning, for example, the scalability of the methods. However, the focus in our research concerning research question 1 was to understand the phenomena found when using prioritization methods in industrial projects. In addition, in order to understand the suitability of the methods for real industrial projects, practitioners’ experience-based opinions are essential.

The threats to **reliability include** the effect of the researchers’ identities to the results. Because the study was this kind of an action research type of a case study, it is evident that it cannot be repeated as such and that it would not have been conducted and reported in exactly the same way if other researcher had conducted it. However, in this case, the question is more about whether another researcher would have provided similar explanations out of the similarish setting. We think that the investigator triangulation used in different phases of the study lowered this risk significantly.

6. Conclusions

6.1 Main contribution and implications

The main contribution of this thesis is that it provides deep understanding about long-term product planning and requirements prioritization in the context of market-driven software product development in practice. Furthermore, the study describes the links of long-term product planning and requirements prioritization to value creation. Our study also analyses and explains the complexity of the requirements prioritization as a phenomenon and the practical challenges involved.

Our results show that current product planning practices in market-driven software product development organizations are at a low-level, focusing on the features of individual products. Plan preparation seems to be restricted to product managers and long-term plans are largely missing. Companies seem to have challenges in drawing together different viewpoints already in the long-term product planning. In particular, combining the business and customer knowledge of product managers with the technical knowledge of project managers and architects, in different phases of product planning, is a key area that seems to be challenging in practice.

In the area of requirements prioritization, our study shows that prioritization practices are informal and dependent on individuals. The priority of a single requirement is based on manifold case-specific aspects. Our study, however, indicates that aspects affecting priorities can be grouped and generalized into-three main points of views: the company's own business, the customers & users, and the implementation. By understanding the role of these aspects and by taking them consciously into account in different levels of product planning, companies could improve their decision-making.

Concerning requirements prioritization challenges, our findings support many of the earlier reported challenges. For example, the number of requirements and the dependencies between them cause serious trouble in practice. Our results, however, also shed new light upon to these challenges. Requirements prioritization is an ambiguous concept that has different things for different people and in different phases of development. In particural, the shift from product level to project level seems to cause misunderstandings in the meaning of the term and scales used. In general, the challenges in requirements prioritization in practice are usully wider than just putting a clearly defined set of requirements in order. Practitioners do not have enough background information upon which to base their prioritization and they cannot incorporate selected aspects in the decision-making process. Practitioners make decisions about priorities mainly without explicitly being aware of which aspects are taken into account and to what extent those aspects are considered.

The study also provided insights for understanding how requirements prioritization methods apply to solving existing challenges in practice in this context. According to our findings, the evaluated requirements prioritization methods have a limited ability to support decision-making in such a complex area as requirements prioritization in market-driven software product development. Our results indicate that current requirements prioritization methods seem to falsely encapsulate the assumption that the value of a release is a sum of its ingredients. According to our study, value is not contained in each requirement as such but instead is created in the process of the user using the system and thus from the interplay of the requirements. Selecting the individually most valuable requirements to be implemented may not lead to maximal product value. This means that the planning focus should be shifted from individual product features to understanding the processes of customers and users and to describing the solutions from the customer's viewpoint as a whole.

6.1.1 Implications for the research

For researchers, this study provides a broad analysis of the practical challenges and characteristics of requirements prioritization and long-term planning in market-driven software development. Our results indicate that requirements prioritization in practice is a broader issue than just comparing a set of requirements at the same level of abstraction with each other, as is assumed in many existing prioritization methods.

Our results provide implications to researchers about what should be taken into account when developing methods and practices for requirements prioritization and long-term planning. For example, the researchers in the field might benefit from understanding that the three main aspects (business, customer&users, and implementation) are used in practice in requirements prioritization. It might be, that the researchers in the area should focus not that much into requirements prioritization method development, but more into trying to solve e.g. the process issues related to combing the three main viewpoints.

The study also analyses the role of long-term product planning and requirements prioritization in value creation and reveals how companies seem to have challenges in drawing different viewpoints together in long-term product planning. According to our study, business, product, project and technology plans, as well as plans for different products are not explicitly connected. Individuals plan for the future, both horizontally (different products) and vertically (business, product, project, technology) in silos. Interestingly, according to our findings, an important success factor is not only the well-known integration of business knowledge into the feature selection, but the technical limitations and possibilities already taken in to account during long-term product planning. All this knowledge about the current state can help researchers to develop practices for solving current challenges. Our study also introduces practices that seem to link requirements prioritization and long-term planning to value creation, and thus providing a basis upon which the other researchers can build.

6.1.2 Implications for practice

From the perspective of market-driven software product development organizations, this thesis introduces practices that seem to support the link from long-term product planning and requirements prioritization to customer value creation. The essence of the suggested practices is in the shift of planning focus from individual product features towards understanding the processes of customers and users and in describing the solutions from the customer viewpoint as a whole. Our study suggests that software product development organizations would need longer-term product planning that is more transparent, more business- and customer-driven, and more cooperative than it is at present.

For practitioners, our results also emphasise that a cross-functional effort is needed already on the product level during long-term planning to be able to thoroughly combine the three main viewpoints (company's own business, customers & users, and implementation) to provide solutions that create customer value. In particular, an area that needs emphasis is combining the business and customer knowledge of product managers with the technical knowledge of project managers and architects, at different phases of product planning.

The solution concept that provides the big picture of the solution and explicates the customer viewpoint seems to make solution planning visible in the organization. Using the solution concept as a planning and communication tool helps to sustain the customer value viewpoint. However, emphasising the solution thinking wider in the organization is needed as well. The practices that seem to support customer value creation and the link to product planning and requirements prioritization include identifying customer segments, establishing direct contacts between the product developers and end users, actively discovering customers' processes and forming a cross-functional team for each main release.

In general, adding intermediate planning levels between business decisions and requirements engineering seems to help to link them together and bring the customer value viewpoint into product planning. Different stakeholders are more able to see what the goals of further releases are, and how well the decisions concerning features match the needs of different customer segments, if the discussion of the product business goals is separated from the R&D resource allocation decisions and detailed feature-level planning (= discussed first). Future priorities are thus discussed not only on the feature level, but also from the viewpoint of high-level targets, which allows different stakeholders to bring their expertise early enough to planning.

In the area of requirements prioritization, the study explains the nature of requirements prioritization in practice. Instead of selecting a 'correct requirements prioritization method, more important issues to be taken into account by an organization in requirements prioritization include: defining the phases in which prioritization is needed; combining the business, customer and technical viewpoints; involving the right stakeholders in different phases; and communicating the priorities and their rationales throug the units of the organization.

6.2 Future directions

The results of this study point to several challenges for the future research.

One important direction for future research is to gain further insights into how the challenges in requirements prioritization and long-term product planning described in this study could be resolved. This study focussed mostly on understanding and analyzing the problem area and only partly on providing solutions. However, since software development practitioners need practical solutions further studies are needed.

A long-term research area that concerns the whole requirements engineering research community still is the continuation of the investigation into why it is so difficult to plan the future development steps of the products in practice. Even though this study provided a reasonable thorough understanding of the current practices and challenges involved, there is still space for more research in providing a deep and thoroughly analyzed understanding of these. On wider research basis, we could better find commonalities between good practices.

Also, one of our future research goals is to acquire further confidence in our results by validating them with new case organizations. We could, for example, collect more good practices from a wide range of other companies to better understand the commonalities between the practices that seem to work best. This could help us form a proposal regarding how to overcome the existing challenges.

In this study, we identified a set of practices that seem to support linking requirements prioritization and long-term product planning to customer value creation. One of our future research goals is to validate these practices with a large number of companies. The relationship between customer value creation and market-driven software development needs more investigation. Therefore, our long-term reach goal is to gain a deep understanding about how long-term planning, requirements engineering and testing can support customer value creation.

References

ACHIMUGU, P., SELAMAT, A., IBRAHIM, R., and MAHRIN, M. N. R. (2014) A systematic literature review of software requirements prioritization research, *Information and Software Technology* (56:6), 6//, 568-585.

ALBRIGHT, R., and KAPPEL, T. (2003) Roadmapping in the Corporation, *Research and Technology Management* (46:2), 31-40.

ANDRIOLE, S. (1998) The Politics of Requirements Management, *IEEE Software* (15:6), 82-84.

AURUM, A., and WOHLIN, C. (2003) The Fundamental Nature of Requirements Engineering Activities as a Decision-Making Process, *Information & Software Technology* (45:14), 94-97.

AURUM, A., and WOHLIN, C. (2007) A Value-Based Approach in Requirements Engineering: Explaining Some of the Fundamental Concepts, in: *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 109-115.

AVISON, D., LAU, F., MYERS, M., and NIELSEN, P. (1999) Action Research, *Communications of the ACM* (42:1), 94-97.

BABAR, M. I., RAMZAN, M., and GHAYYUR, S. A. K. (2011) Challenges and Future Trends in Software Requirements Prioritization, *2011 International Conference on Computer Networks and Information Technology (ICCNIT)*), 11-13 July 2011, 319-324.

BAGNALL, A. J., RAYWARD-SMITH, V. J., and WHITTLEY, I. M. (2001) The Next Release Problem, *Information and Software Technology* (43:14), 883-890.

BAKALOVA, Z., DANEVA, M., HERRMANN, A., and WIERINGA, R. (2011) Agile Requirements Prioritization: What Happens in Practice and What Is Described in Literature, in: *17th International Working Conference on Requirements Engineering - Foundation for Software Quality (REFSQ 2011)*, Essen, Germany, pp. 181-195.

BARNEY, S., AURUM, A., and WOHLIN, C. (2008) A Product Management Challenge: Creating Software Product Value through Requirements Selection, *Journal of Systems Architecture* (54:6), 576-593.

BARNEY, S., WOHLIN, C., GANGLAN, H., and AURUM, A. (2009) Creating Software Product Value in China, *IEEE Software* (26:4), 84-90.

BAVANI, R. (2010) Global Software Engineering: Challenges in Customer Value Creation, Proceedings of Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on, pp. 119-122.

BEARDEN, W., and TEEL, J. (1983) Selected Determinants of Consumer Satisfaction and complaint Reports., *Journal of Marketing Research (JMR)* (20:1), 21-28.

BECK, K. (1999) Embracing Change with Extreme Programming, *Computer* (32:10), 70-77.

BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J., MARICK, B., MARTIN, R. C., MELLOR, S., SCHWABER, K., SUTHERLAND, J., and THOMAS, D. (2001) Manifesto for Agile Software Development, http://www.agilemanifesto.org/.

BENESTAD, H. C., and HANNAY, J. E. (2011) A Comparison of Model-Based and Judgment-Based Release Planning in Incremental Software Projects, in: *33rd International Conference on Software Engineering (ICSE)*, Honolulu, Hawaii, pp. 766-775.

BERANDER, P. (2007) Evolving Prioritization for Software Product Management, in: *School of Engineering, Department of Sytems and Software Engineering*, Blekinge Institute of Technology.

BERANDER, P., and ANDREWS, A. (2005) Requirements Prioritization, in: *Engineering and Managing Software Requirements,* A. Aurum and C. Wohlin (eds.), Springer Verlag, Berlin, Germany, pp. 69-94.

BERANDER, P., and JÖNSSON, P. (2006a) Hierarchical Cumulative Voting (HCV) - Prioritization of Requirements in Hierarchies, *International Journal of Software Engineering and Knowledge Engineering* (16:6 (Special Issue on Requirements Engineering Decision Support)), 819-850.

BERANDER, P., KHAN, K. A., and LEHTOLA, L. (2006b) Towards a Research Framework on Requirements Prioritization, in: *Sixth Conference on Software Engineering Research and Practise in Sweden (SERPS'06)*, Umea, Sweden, pp. 39-48.

BERANDER, P., and SVAHNBERG, M. (2009) Evaluating Two Ways of Calculating Priorities in Requirements Hierarchies - An Experiment on Hierarchical Cumulative Voting, *Journal of Systems and Software* (82:5), 836-850.

BJARNASON, E., WNUK, K., and REGNELL, B. (2010) Overscoping: Reasons and Consequences - A Case Study on Decision Making in Software Product Management, in: *2010 Fourth International Workshop on Software Product Management (IWSPM)*, Sydney, Australia, pp. 30-39.

BOEHM, B. (2003) Value-Based Software Engineering, *ACM Software Engineering Notes* (28:2), 1-12.

BOEHM, B., EGYED, A., KWAN, J., PORT, D., SHAH, A., and MADACHY, R. (1998) Using the WinWin Spiral Model: A Case Study, *Computer* (31:7), 33-44.

BOEHM, B., GRUNBACHER, P., and BRIGGS, R. O. (2001) Developing Groupware for Requirements Negotiation: Lessons Learned, *IEEE Software* (18:3), 46-55.

BOEHM, B., and LI GUO, H. (2003) Value-based software engineering: a case study, *Computer* (36:3), 33-41.

BOLAND, R. (1985) Phenomenology: A preferred approach to research on information systems, in: *Research Methods in Information Systems*, pp. 193-201.

BOSCH, J. (2000) *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach* ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, p. 354.

BUBENKO, J. (1995) Challenges in Requirements Engineering, in: *IEEE International Symposium on Requirements Engineering*, IEEE Computer Society Press, pp. 160-162.

CARLSHAMRE, P. (2002) Release Planning in Market-Driven Software Product Development - Provoking an Understanding, *Requirements Engineering* (7:3), 13.

CARLSHAMRE, P., SANDAHL, K., LINDVALL, M., REGNELL, B., and NATT OCH DAG, J. (2001) An Industrial Survey of Requirements Interdependencies in Software Product Release Planning, Proceedings of Fifth IEEE International Symposium on Requirements Engineering, pp. 84-91.

CARMEL, E., and SAWYER, S. (1998) Packaged Software Development Teams: What Makes Them Different?, *Information Technology & People* (11:1), 7-19.

CORMEN, T. H., LEISERSON, C. E., and RIVEST, R. L. (2006) *Introduction to Algorithms*, (Third ed.) McGraw Hill, New York, p. 1321.

CUSUMANO, M. A. (2004) *The Business of Software - What Every Manager, Programmer and Entrepreneur Must Know How to Thrive And Survive in Good Times and Bad* Free Press, New York, p. 352.

DAMIAN, D. (2001) Negotiation behavior and group interaction in face-to-face and distributed requirements negotiations: four case studies, in: *Australian Workshop on Requirements Engineering*, Sydney, Australia, pp. 22-31.

DAMIAN, D., and ZOWGHI, D. (2003) RE Challenges in Multi-Site Software Development Organizations, *Requirements Engineering* (8:3), 149-160.

DAVIS, A. (1993) *Software Requirements - Objects, Functionas, & States*, (Revision ed.) Prentice Hall, Upper Saddle River, New Jersey, p. 521.

DE GREGORIO, G. (2000) Technology Management via a Set of Dynamically Linked Roadmaps, in: *IEEE International Engineering Management Conference* IEEE Engineering Management Society, pp. 184-190.

DENNE, M., and CLELAND-HUANG, J. (2004) The Incremental Funding Method: Data-Driven Software Development, *IEEE Software* (21:3), 39-47.

DUBÉ, L., and PARÉ, G. (2003) Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations, *MIS Quarterly* (27:4), 597-636.

DVER, A. S. (2003) *Software product management essentials - a practical guide for small and midsized companies* Meghan Kiffer Pr, Tampa, Florida, p. 202.

EBERT, C. (2005) Requirements BEFORE the Requirements: Understanding the Upstream Impacts, in: *13th IEEE International Conference on Requirements Engineering*, IEEE Computer Society Press, pp. 117-124.

EBERT, C. (2006) The Impacts of Software Product Management, *The Journal of Systems and Software* (80:6), 850-861.

EBERT, C. (2009) Software Product Management, *Crosstalk (*http://www.crosstalkonline.org*)* (2009:1), 15-19.

EISENHARDT, K. M. (1989) Building Theories from Case Study Research, *Academy of Management Review* (14:4), 532-550.

EJNIOUI, A., OTERO, C. E., and QURESHI, A. A. (2012) Software Requirement Prioritization Using Fuzzy Multi-Attribute Decision Making, in: *IEEE Conference on Open Systems (ICOS)*, Kuala Lumbur, pp. 1-6.

FAVARO, J. (2002) Managing Requirements for Business Value, *IEEE Software* (19:2), 15-17.

FELLOWS, L., and HOOKS, I. (1998) A Case for Priority Classifying Requirements, in: *Eight Annual International Symposium on Systems Engineering*, International Council on Systems Engineering, Seattle, Washingon, pp. 1-8.

FERNANDES, M. C., ALENCAR, A. J., SCHMITZ, E. A., ALVES, C. H., and FERREIRA, A. L. (2008) A Multicriteria Approach to the XP Release Plan that Maximizes Business Performance in Uncertain Environments, in: *Systems and Information Engineering Design Symposium (SIEDS 2008)* pp. 184-189.

FIRESMITH, D. (2004) Prioritizing Requirements, *Journal of Object Technology* (3:8), 35-47.

FORNELL, C. (1992) A National Customer Satisfaction Barometer: The Swedish Experience, *Journal of Marketing* (56:6), 6-21.

GALVIN, R. (2004) Roadmapping--A Practitioner's Update, *Technological Forecasting and Social Change* (71:1-2), 101-103.

GARDIAL, S., CLEMONS, D., WOODRUFF, R., SCHUMANN, D., and BURNS, M. (1994) Comparing Consumers' Recall of Prepurchase and Postpurchase Product Evaluation Experiences, *Journal of Consumer Research* (20:4), 548-560.

GILB, T. (2005) *Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage* Butterworth-Heinemann, p. 480.

GORCHELS, L. (2000) *The Product Manager's Handbook - the Complete Product Management Resource*, (2nd ed.) NTC Business books, Illinois, USA.

GREER, D., and RUHE, G. (2004) Software Release Planning: an Evolutionary and Iterative Approach, *Information and Software Technology* (46:4), 243-253.

GROENVELD, P. (1997) Roadmapping Intergates Business and Technology, *Research and Technology Management* (40:5), 48-55.

GRÖNROOS, C. (2007) *Service Management and Marketing: Customer Management in Service Competition*, (third ed.), p. 496.

HAAR, J. W. V. D., KEMP, R. G. M., and OMTA, O. (2001) Creating Value that Cannot Be Copied, *Industrial Marketing Management* (30:8), 627-636.

HARWELL, R., ASLAKSEN, E., HOOKS, I., MENGOT, R., and PTACK, K. (1993) What is a Requirement?, in: *Third International Symposium of the NCOSE*, pp. 17-24.

HEIKKILÄ, V., JADALLAH, A., RAUTIAINEN, K., and RUHE, G. (2010a) Rigorous Support for Flexible Planning of Product Releases - A Stakeholder-Centric Approach and Its Initial Evaluation, in: *43rd Hawaii International Conference on System Sciences (HICSS)* Honolulu, Hawaii, pp. 1-10.

HEIKKILÄ, V., RAUTIAINEN, K., and JANSEN, S. (2010b) A Revelatory Case Study on Scaling Agile Release Planning, in: *36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)* LIlle, France, pp. 289-296.

HERRMANN, A., and DANEVA, M. (2008) Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research, in: *16th IEEE International Requirements Engineering Conference (RE '08)* Barcelona, Spain, pp. 125-134.

HERRMANN, A., and PAECH, B. (2006) Benefit Estimation of Requirements Based on a Utility Function, in: *12th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'06)*, pp. 249-250.

HOCH, D., ROEDING, C., PURKERT, G., LINDNER, S., and MÜLLER, R. (1999) *Secrets of Software Success: Management Insights from 100 Software Firms Around the World* Harvard Business Review Press, Boston, p. 312.

HOLBROOK, M. B. (2006) Consumption Experience, Customer Value, and Subjective Personal Introspection: An Illustrative Photographic Essay, *Journal of Business Research* (59:6), 714-725.

HUBER, F., HERRMANN, A., and MORGAN, R. E. (2001) Gaining Competitive Advantage through Customer Value Oriented Management, *Journal of Consumer Marketing* (18:1), 41-53.

IQBAL, M. A., ZAIDI, A. M., and MURTAZA, S. (2010) A New Requirement Prioritization Model for Market Driven Products Using Analytical Hierarchical Process, in: *International Conference on Data Storage and Data Engineering (DSDE)*, Kerala, India, pp. 142-149.

JACKSON, M. (1995) *Requirements and Specifications - A Lexicon of Practice, Principles and Prejudices* Addison-Wesley Professional, Wokingham, p. 228.

JANTUNEN, S., and SMOLANDER, K. (2006) Challenges of Knowledge and Collaboration in Roadmapping, in: *International Workshop on Software Product Management (IWSPM '06)* pp. 19-26.

JUNG, H. (1998) Optimizing Value and Cost in Requirements Analysis, *IEEE Software* (15:4), 74-78.

KANO, N., SERAKU, N., TAKAHASHI, F., and TSUJI, S. (1984) Attractive Quality and Must-be Quality, *Journal of Japanise Society for Quality Control* (14:2), 39-48.

KAPPEL, T. (2001) Perspectives on Roadmaps: How Organisations Talk about the Future, *IEEE Engineering Management Review* (29:3), 36-48.

KARLSSON, J. (1996) Software Requirements Prioritizing, in: *2nd IEEE International Conference on Requirements Engineering (ICRE1996)*, Colorado Springs, CO, pp. 110-116.

KARLSSON, J., OISSON, S., and RYAN, K. (1997a) Improved Practical Support for Large-Scale Requirements Prioritising, *Requirements Engineering* (2:1), 51-60.

KARLSSON, J., and RYAN, K. (1997b) A Cost-Value Approach for Prioritizing Requirements, *IEEE Software* (14:5), 67-74.

KARLSSON, J., WOHLIN, C., and REGNELL, B. (1998) An Evaluation of Methods for Prioritizing Software Requirements, *Information & Software Technology* (39:14-15), 939-947.

KARLSSON, L., BERANDER, P., REGNELL, B., and WOHLIN, C. (2004) Requirements Prioritisation: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning, in: *8th Internation Conference on Empirical Assessment in Software Engineering (EASE 2004) Workshop - 26th International Conference on Software Engineering*, Edinbourgh, Scotland, UK, pp. 145-154.

KARLSSON, L., DAHLSTEDT, G., NATT OCH DAG, J., REGNELL, B., and PERSSON, A. (2002) Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study, in: *8th International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'02)*, Essen, Germany, pp. 37-49.

KARLSSON, L., DAHLSTEDT, Å. G., REGNELL, B., DAG, J. N. O., and PERSSON, A. (2007a) Requirements Engineering Challenges in Market-Driven Software Development - An Interview Study with Practitioners, *Information & Software Technology* (49:6), 588-604.

KARLSSON, L., and REGNELL, B. (2005) Comparing Ordinal and Ratio Scale Data in Requirements Prioritisation, in: *3rd International Workshop on Comparative Evaluation in Requirements Engineering (CERE'05)*, Paris, France, pp. 1-13.

KARLSSON, L., THELIN, T., REGNELL, B., BERANDER, P., and WOHLIN, C. (2007b) Pair-Wise Comparisons versus Planning Game Partitioning - Experiments on Requirements Prioritisation Techniques, *Empirical Software Engineering* (12:1), 3-33.

KHALIFA, A. S. (2004) Customer Value: a Review of Recent Literature and an Integrative Configuration, *Management Decision* (42:5), 645-666.

KHURUM, M., GORSCHEK, T., and WILSON, M. (2013) The software value map — an exhaustive collection of value aspects for the development of software intensive products, *Journal of Software: Evolution and Process* (25:7), 711-741.

KHURUM, M., UPPALAPATI, N., and VEERAMACHANENI, R. C. (2012) Software Requirements Triage and Selection: State-of-the-Art and State-of-Practice, in: *19th Asia-Pacific Software Engineering Conference (APSEC)* Hong Kong, China, pp. 416-421.

KOMSSI, M., KAUPPINEN, M., TOHONEN, H., LEHTOLA, L., and DAVIS, A. M. (2011) Integrating Analysis of Customers' Processes into Roadmapping: The Value-Creation Perspective, in: *19th IEEE International Requirements Engineering Conference (RE)*, Trento, Italy, pp. 57-66.

KOTONYA, G., and SOMMERVILLE, I. (1998) *Requirements Engineering - Processes and Techniques* Wiley, Chichester, p. 294.

KUKREJA, N., PAYYAVULA, S. S., BOEHM, B., and PADMANABHUNI, S. (2012) Selecting an Appropriate Framework for Value-Based Requirements Prioritization, in: *20th IEEE International Requirements Engineering Conference (RE'12)*, Chicago, Illinois, USA, pp. 303-308.

LAUESEN, S. (2002) *Software Requirements: Styles & Techniques* Addison-Wesley Professional, Harlow, England, p. 608.

LAURENT, P., CLELAND-HUANG, J., and CHUAN, D. (2007) Towards Automated Requirements Triage, in: *15th IEEE International Requirements Engineering Conference (RE '07)*, Delhi, India, pp. 131-140.

LEFFINGWELL, D., and WIDRIG, D. (2000) *Managing Software Requirements - A Unified Approach* Addison-Wesley, p. 528.

LEFFINGWELL, D., and WIDRIG, D. (2003) *Managing Software Requirements: A Use-Case Approach*, (Second ed.) Addison-Wesley, Boston, p. 544.

LINCOLN, Y. S., and GUBA, E. G. (1985) *Naturalistic Inquiry* SAGE Publications.

LUBARS, M., POTTS, C., and RICHTER, C. (1993) A Review of the State of the Practice in Requirements Modeling, in: *IEEE Symposium on Requirements Engineering (RE'93)*, IEEE Computer Society Press, San Diego, California, pp. 2-14.

MACMILLAN, I. C., and MCGRATH, R. G. (1997) Discovering New Points of Differentiation, *Harvard Business Review* (75:4), 133-145.

MILES, L. D. (1961) *Techniques of Value Analysis and Engineering* McGraw Hill, New York, p. 351.

MOISIADIS, F. (2002) The Fundamentals of Prioritizing Requirements, in: *Test and Automation Conference (SETE2002)*, Sydney, Australia, pp. 1-12.

MURTAZAEV, A., SUNGWON, K., JONGMOON, B., and JIHYUN, L. (2010) An Approach to Defining a Value-Based Software Development Process, Proceedings of Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on, pp. 690-695.

MYERS, M. D. (2009) *Qualitative Research in Business & Management*, (1st ed.) SAGE publications Ltd, p. 296.

NAMBISAN, S. (2001) Why Service Businesses are not Product Businesses, *MIT Sloan Management Review* (42:4), 72-80.

NAUMANN, E. (1995) *Creating Customer Value: The Path to Sustainable Competitve Advantage* Thomson Executive Press Cincinnati, Ohio.

NGO-THE, A., and RUHE, G. N. (2005) Decision Support in Requirements Engineering, in: *Engineering and Managing Software Requirements,* A.a.W. Aurum, C (ed.), Springer Verlag, Berlin, Germany, pp. 267-286.

NORMANN, R., and RAMÍREZ, R. (1993) From Value Chain to Value Constellation: Designing Interactive Strategy, *Harvard Business Review* (71:4), 65-77.

NOVORITA, R., and GRUBE, G. (1996) Benefits of Structured Requirements Methods for Market-Based Enterprises, in: *Sixth Annual International Symposium on Systems Engineering: Practices and Tools (INCOSE)*, Boston, Mass., USA.

NUSEIBEH, B., and EASTERBROOK, S. (2000) Requirements Engineering: A Roadmap, in: *22nd International Conference on Software Engineering,* A. Finkelstein (ed.), ACM Press, pp. 35-46.

ORLIKOWSKI, W. J., and BAROUDI, J. J. (1991) Studying Information Technology in Organizations: Research Approaches and Assumptions, *Information Systems Research* (2:1), 3, 1-28.

PAECH, B., and AURUM, A. (2008) Boundary Objects for Value-based Requirements Engineering, in: *Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems*, Marburg, Germany, pp. 11-24.

PARASURAMAN, A. (1997) Reflections on Gaining Competitive Advantage through Customer Value, *Journal of the Academy of Marketing Science* (25:2), 154-161.

PARK, J.-W., PORT, D., and BOEHM, B. (1999) Supporting Distributed Collaborative Prioritization, in: *Sixth Asia Pacific Software Engineering Conference (APSEC '99)* pp. 560-563.

PATTON, I. (2008) Ambiguous Business Value Harms Software Products, *Software, IEEE* (25:1), 50-51.

PATTON, M. Q. (2002) *Qualitative Research and Evaluation Methods*, (3rd ed.) Sage Publications, Thousand Oaks, CA., p. 688.

PENNY, D. A. (2002a) An Estimation-Based Management Framework for Enhancive Maintenance in Commercial Software Products, in: *IEEE International Conference on Software Maintenance*, pp. 122-130.

PENNY, D. A. (2002b) Release Planning, in: *International Conference on Software Maintenance (ICSM'02)*, p. 1.

PERGHER, M., and ROSSI, B. (2013) Requirements Prioritization in Software Engineering: A Systematic Mapping Study, in: *IEEE Third International Workshop on Empirical Requirements Engineering (EmpiRE)* Rio de Janeiro, pp. 40-44.

PERINI, A., RICCA, F., and SUSI, A. (2009) Tool-Supported Requirements Prioritization: Comparing the AHP and CBRank Methods, *Information and Software Technology* (51:6), 1021-1032.

PETERSEN, K., and WOHLIN, C. (2009) A Comparison of Issues and Advantages in Agile and Incremental Development between State of the Art and an Industrial Case, *Journal of Systems and Software* (82:9), 1479-1490.

PHAAL, R., FARRUKH, C. J. P., MITCHELL, R., and PROBERT, D. R. (2003) Starting up Roadmapping Fast, *Research Technology Management* (46:2), 52-58.

PHAAL, R., FARRUKH, C. J. P., and PROBERT, D. R. (2001) Characterisation of Technology Roadmaps: Purpose and Format., in: *Portland International Conference on Management of Engineering and Technology (PICMET'01)*, pp. 367-374.

PHAAL, R., FARRUKH, C. J. P., and PROBERT, D. R. (2004) Technology Roadmapping - A Planning Framework for Evolution and Revolution, *Technological Forecasting & Social Change* (71:1-2), 5-26.

PITANGUEIRA, A., MACIEL, R., DE OLIVEIRA BARROS, M., and ANDRADE, A. (2013) A Systematic Review of Software Requirements Selection and Prioritization Using SBSE Approaches, in: *Search Based Software Engineering,* G. Ruhe and Y. Zhang (eds.), Springer Berlin Heidelberg, pp. 188-208.

PORTER, M. (2004) *Competitive Advantage: Creating and Sustaining Superior Performance*, (First Free Press Export Edition ed.) Free Press, New York, p. 592.

POTTS, C. (1993) Software-Engineering Research Revisited, *IEEE Software* (10:5), 19-28.

POWELL, R. A., and SINGLE, H. M. (1996) Focus groups., *International Journal for Qulity in Health Care* (8:5), 499-504.

RACHEVA, Z., DANEVA, M., and SIKKEL, K. (2009) Value Creation by Agile Projects: Methodology or Mystery?, in: *Product-Focused Software Process Improvement,* F. Bomarius, M. Oivo, P. Jaring and P. Abrahamsson (eds.), Springer Berlin Heidelberg, pp. 141-155.

RACHEVA, Z., DANEVA, M., SIKKEL, K., and BUGLIONE, L. (2010a) Business Value Is Not Only Dollars – Results from Case Study Research on Agile Software Projects, in: *Product-Focused Software Process Improvement,* M. Ali Babar, M. Vierimaa and M. Oivo (eds.), Springer Berlin Heidelberg, pp. 131-145.

RACHEVA, Z., DANEVA, M., SIKKEL, K., WIERINGA, R., and HERRMANN, A. (2010b) Do we Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study, in: *18th International Requirements Engineering Conference*, IEEE Computer Society Press, Sydney, Australia, pp. 147-156.

RAUTIAINEN, K., LASSENIUS, C., VÄHÄNIITTY, J., VANHANEN, J., and PYHÄJÄRVI, M. (2002) A Tentative Framework for Managing Software Product Development in Small Companies, in: *Thirty-Third Annual Hawaii International Conference on System Sciences*, Honolulu, Hawaii, pp. 251-260.

RAUTIAINEN, K., VUORNOS, L., and LASSENIUS, C. (2003) An Experience in Integrating Strategic Product Planning and Agile Software Development Practices, in: *2003 International Symposium on Empirical Software Engineering*, Los Alamitos, CA, pp. 28-37.

REGNELL, B., BEREMARK, P., and EKLUNDH, O. (1998) A Market-Driven Requirements Engineering Process: Results from an Industrial Process Improvement Programme, *Requirements Engineering* (3:2), 121-129.

REGNELL, B., and BRINKKEMPER, S. (2005) Market-Driven Requirements Engineering for Software Products, in: *Engineering and Managing Software Requirements,* A. Aurum and C. Wohlin (eds.), Springer, pp. 287-308.

REGNELL, B., HÖST, M., NATT OCH DAG, J., BEREMARK, P., and HJELM, T. (2001) An Industrial Case Study on Distributed Prioritization in Market- Driven Requirements Engineering for Packaged Software, *Requirements Engineering* (6:1), 51-62.

REGNELL, B., and KUCHCINSKI, K. (2011) Exploring Software Product Management Decision Problems with Constraint Solving - Opportunities for Prioritization and Release Planning, in: *Fifth International Workshop on Software Product Management (IWSPM)* Trento, Italy, pp. 47-56.

REGNELL, B., SVENSSON, R. B., and OLSSON, T. (2008) Supporting Roadmapping of Quality Requirements, *Software, IEEE* (25:2), 42-47.

ROBSON, C. (2002) *Real word research* Blackwell.

ROYCE, W. W. (1970) Managing the Development of Large Software Systems: Concepts and Techniques, in: *9th international conference on Software Engineering (ICSE'87)*, IEEE, Wescon, pp. 328-338.

RUHE, G. (2010) *Product Release Planning: Methods, Tools and Applications* Auerbach Publications, Boca Raton, FL, p. 339.

RUHE, G., EBERLEIN, A., and PFAHL, D. (2002) Quantitative WinWin - A New Method for Decision Support in Requirements Negotiation, in: *14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02)*, Ischia, Italy, pp. 159-166.

RUHE, G., EBERLEIN, A., and PFAHL, D. (2003) Trade-off Analysis for Requirements Selection, *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)* (13:4), 345-366.

RUNESON, P., and HÖST, M. (2009) Guidelines for conducting and reporting case study research in software engineering, *Empirical Software Engineering* (14:2), 2009/04/01, 131-164.

RUNESON, P., HÖST, M., AUSTIN, R., and REGNELL, B. (2012) *Case Study Research in Software Engineering: Guidelines and Examples* Wiley, p. 256.

SAATY, T. L. (1980) *The Analytical Hierarchy Process: : Planning, Priority Setting, Resource Allocation (Decision Making Series)* McGraw-Hill, New York, p. 287.

SAWYER, P. (2000) Packaged Software: Challenges for Requirements Engineering, in: *6th International Workshop on Requirements Engineering: Foundation of Software Quality*, Stockholm, Sweden, pp. 137-142.

SCHWABER, K. (2004) *Agile Project Management with Scrum* Microsoft Press, Redmond, WA.

SEAMAN, C. B. (1999) Qualitative Methods in Empirical Studies of Software Engineering, *IEEE Transactions on Software Engineering* (25:4), 557-572.

SHARMA, D., AURUM, A., and PAECH, B. (2008) Business Value through Product Line Engineering - A Case Study, in: *34th Euromicro Conference on Software Engineering and Advanced Applications (SEAA '08)*, Parma, Italy, pp. 167-174.

SIDDIQI, J., and SHEKARAN, C. (1996) Requirements Engineering: The Emerging Wisdom, *IEEE Software* (13:2), 15-19.

SIMONS, H. (2009) *Case Study Research in Practice.* SAGE Publications, Ltd, London.

SIVZATTIAN, S., and NUSEIBEH, B. (2001) Linking the Selection of Requirements to Market Value: A Portfolio-Based Approach, in: *7th International Workshop on Requirements Engineering: Foundation for Software Quality*, Interlaken, Switzerland, pp. 202-213.

SLATER, S. F. (1997) Developing a Customer Value-Based Theory of the Firm, *Journal of the Academy of Marketing Science* (25:2), 162-167.

SMITH, J. B., and COLGATE, M. (2007) Customer Value Creation: A Practical Framework, *Journal of Marketing Theory and Practice.* (15:1), 7-23.

SOBCZAK, A., and BERRY, D. M. (2007) Distributed Priority Ranking of Strategic Preliminary Requirements for Management Information Systems in Economic Organizations, *Information and Software Technology* (49:9-10), 960-984.

SOMMERVILLE, I. (2010) *Software Engineering*, (Ninth ed.) Addison-Wesley Publishing Company, Wokingham, p. 792.

STRINGER, E. T. (1999) *Action Research* (2 ed.) Sage publications, p. 279.

SVAHNBERG, M., GORSCHEK, T., FELDT, R., TORKAR, R., SALEEM, S. B., and SHAFIQUE, M. U. (2010) A Systematic Review on Strategic Release Planning Models, *Information and Software Technology* (52:3), 237-248.

SVENSSON, R. B., GORSCHEK, T., REGNELL, B., TORKAR, R., SHAHROKNI, A., FELDT, R., and AURUM, A. (2011) Prioritization of Quality Requirements: State of Practice in Eleven Companies, in: *19th IEEE International Requirements Engineering Conference (RE)*, Trento, Italy, pp. 69-78.

SVENSSON, R. B., SPROCKEL, Y., REGNELL, B., and BRINKKEMPER, S. (2010) Cost and Benefit Analysis of Quality Requirements in Competitive Software Product Management: A Case Study on the QUPER Model, in: *Fourth International Workshop on Software Product Management (IWSPM)*, Sydney, Australia, pp. 40-48.

ULAGA, W. (2003) Capturing Value Creation in Business Relationships: A Customer Perspective, *Industrial Marketing Management* (32:8), 677-693.

ULAGA, W., and CHACOUR, S. (2001) Measuring Customer-Perceived Value in Business Markets: A Prerequisite for Marketing Strategy Development and Implementation, *Industrial Marketing Management* (30:6), 525-540.

VAN DE WEERD, I., BRINKKEMPER, S., NIEUWENHUIS, R., VERSENDAAL, J., and BIJLSMA, L. (2006) Towards a Reference Framework for Software Product Management, Proceedings of Requirements Engineering, 14th IEEE International Conference, pp. 319-322.

VAN DEN AKKER, M., BRINKKEMPER, S., DIEPEN, G., and VERSENDAAL, J. (2008) Software Product Release Planning through Optimization and What-If Analysis, *Information and softwareTechnology* (50:1-2), 101-111.

VARGO, S. L., and LUSCH, R. F. (2004) Evolving to a New Dominant Logic for Marketing, *Journal of Marketing* (68:1), 1-17.

VATANANAN, R. S., and GERDSRI, N. (2010) The Current State of Technology Roadmapping (TRM) Research and Practice, in: *Technology Management for Global Economic Growth (PICMET)*, Phuket, Thailand, pp. 1-10.

VOOLA, P., and BABU, A. V. (2013) Comparison of Requirements Prioritization Techniques Employing Different Scales of Measurement, *SIGSOFT Software Engineering Notes* (38:4), 1-10.

VÄHÄNIITTY, J. (2005) A Tentative Framework for Connecting Long-Term Business and Product Planning with Iterative & Incremental Software Product Development, in: *7th International Workshop on Economics-Driven Software Engineering Research*, ACM, St. Louis, Missouri, pp. 1-4.

VÄHÄNIITTY, J., LASSENIUS, C., and RAUTIAINEN, K. (2002) An Approach to Product Roadmapping in Small Software Product Businesses, in: *Quality Connection - 7th European Conference on Software Quality - Conference Notes*, Center for Excellence Finland, Helsinki, Finland, pp. 12-13.

VÄHÄNIITTY, J., LASSENIUS, C., RAUTIAINEN, K., and PEKKANEN, P. (2009) Long-Term Planning of Development Efforts by Roadmapping: A Model and Experiences from Small Software Companies, in: *35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA '09)* Patras, Greece, pp. 300-305.

WIEGERS, K. (1999) *Software Requirements* Microsoft Press, p. 350.

WIEGERS, K. (2003) *Software Requirements, Second Edition* Microsoft Press, p. 544.

WIERINGA, R., and DANEVA, M. (2015) Six strategies for generalizing software engineering theories, *Science of Computer Programming* (101:0), 4/1/, 136-152.

WIERINGA, R. J. (2014) *Design Science Methodology for Information Systems and Software Engineering* Springer-Verlag, Berlin Heideberg, p. 332.

WOHLIN, C., and AURUM, A. (2005a) Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study, in: *Value-Based Software Engineering,* S. Biffl, A. Aurum, B. Boehm, H. Erdogan and P. Gr√°nbacher (eds.), Springer Verlag.

WOHLIN, C., and AURUM, A. (2005b) What is Important when Deciding to Include a Requirement in a Project or Release?, in: *The Fourth International Symposium on Empirical Software Engineering Conference*, Noosa Heads, Queensland, Australia, pp. 246-255.

WOODRUFF, R. B. (1997) Customer Value: The Next Source for Competitive Advantage, *Journal of the Academy of Marketing Science* (25:2), 139-153.

WOODRUFF, R. B., and GARDIAL, S. F. (1996) *Know your Customer: New approaches to Understanding Customer Value and Satisfaction* Blackwell Publishers, Cambridge, p. 360.

YEH, A. C. (1992) Requirements Engineering Support Technique (request) - A Market Driven Requirements Management Process, in: *Second Symposium of Quality Software Development Tools*, New Orleans, USA, pp. 211-223.

YIN, R. K. (1994) *Case Study Research: Design and Methods*, (Second ed.) SAGE Publications, London, p. 194.

YIN, R. K. (2009) *Case Study Research: Design and Methods* SAGE Publications.

Aalto-DD 217/2015

Creating value for different customer segments is essential to the business of a company. Thus, software product development companies' ability to implement the most valuable requirements in their products has been seen as critical.

This thesis investigates the current state of long-term product planning and requirements prioritization, and their linkages to customer value creation in 7 Finnish software product development companies.

As a result, a systematic analysis of long-term product planning and requirements prioritization activities and challenges is provided. The thesis also proposes a set of practices that support the link from long-term product planning and requirements prioritization to customer value creation.

9 789526 065908

ISBN 978-952-60-6590-8 (printed)
ISBN 978-952-60-6591-5 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS