HELSINKI UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering

LAURA LEHTOLA

Providing value by prioritizing requirements throughout software product development

State of practice and suitability of prioritization methods

Licentiate thesis submitted for official examination for the degree of Licentiate in Technology.

Espoo, 05.09.2006

Instructor: Marjo Kauppinen

Supervisor: Tomi Männistö

HELSINKI UNIVERSITY OF TECHNOLOGY **ABSTRACT OF LICENTIATE THESIS**

Department of Computer Science and Engineering

Author
Laura Lehtola

Date
5.9.2006

Pages
53 + publications

Title of thesis

Providing value by prioritizing requirements throughout software product development - State of practice and suitability of prioritization methods

Professorship	Professorship Code
Software engineering	T-76

Supervisor
Tomi Männistö

Instructor
Marjo Kauppinen

For a software company product development is an investment that should provide value for customers and end-users. Providing value for these stakeholders is a lifeline for the sales of the product and via that to the business of the software company.

In order to provide value, a company must be able to select the most valuable requirements to be implemented in each product release. In the software product business these decisions cannot be negotiated with just one customer. According to earlier studies, the prioritization of requirements is not an easy task. Extensive description concerning the challenges involved and analysis of their root causes has, however, not been introduced. In addition, the literature offers methods for prioritization, but these methods are not widely used.

In this research, we investigated the existing mechanisms that software companies operating in the product business use in order to prioritize features and requirements during product development. In addition, we clarified the challenges involved in prioritization and in practices that aim to address these challenges. This was done in order to understand the problem area better and through that to provide a basis for improving the existing practices. In addition, we evaluated two requirements prioritization methods from the requirements engineering literature in software development projects to get information about their suitability for market-driven software development.

In practice, requirements are prioritized in many phases of product development. On the product management level, requirements are prioritized mainly from the business viewpoint and allocated to a few forthcoming releases so that the time horizon for future planning is open. The steps for the near future are planned in more detail and those for the remote future in less detail. On the project level, requirements are again prioritized into an indicative implementation order based on e.g. the skills available and the maturity of the requirements. The practices used are mainly informal, but most often involve evaluating the potential business value of requirements versus their implementation costs. In addition, companies aim to utilize their product development resources evenly as a function of time.

Prioritization involves manifold challenges, which include, for example, identifying the business potential of a requirement, the fact that features affecting priorities are case-specific, and difficulties in involving secondary stakeholders in decision-making. Another central challenge is the description and communication of priority information through the organization. Prioritization methods may provide help in prioritizing a limited set of requirements, but organizing a group of mature requirements in an order is just a part of the challenge.

Keywords:
requirements engineering, requirements prioritization, release planning, roadmapping

TEKNILLINEN KORKEAKOULU Tietotekniikan osasto	**LISENSIAATINTUTKIMUKSEN TIIVISTELMÄ**

Tekijä Laura Lehtola	Päiväys 5.9.2006
	Sivumäärä 53 + julkaisut

Työn nimi

Arvontuotto ohjelmistotuotekehityksessä toistuvan vaatimusten priorisoinnin avulla –

Nykytila ja priorisointimenetelmien soveltuvuus

Professuuri Ohjelmistoliiketoiminta ja –tuotanto	Professuurin koodi T-76

Työn valvoja

Tomi Männistö

Työn ohjaaja

Marjo Kauppinen

Ohjelmistotuoteyritykselle tuotekehitys on investointi, jonka tarkoituksena on tuottaa mahdollisimman paljon lisäarvoa asiakkaille ja loppukäyttäjille. Arvon tuottaminen näille sidosryhmille on elinehto tuotteen myynnille ja sitä kautta yrityksen omalle liiketoiminnalle.

Arvontuoton onnistumiseksi yrityksen on kyettävä valitsemaan kuhunkin julkaisuun ominaisuudet, joiden avulla tuotettava lisäarvo on mahdollisimman suuri. Ohjelmistotuoteliiketoiminnassa, toisin kuin projektiliiketoiminnassa, ominaisuuspäätöksistä ei voida neuvotella vain yhden asiakkaan kanssa. Aikaisemman tutkimuksen perusteella ominaisuuspäätösten tekeminen on vaikea ja haasteellinen tehtävä. Kattavaa kuvausta haasteista ja analyysiä niiden aiheuttajista ei kuitenkaan ole esitetty. Kirjallisuudessa tarjotaan myös joitakin priorisointimenetelmiä yritysten käyttöön, mutta nämä menetelmät eivät ole laajalti käytössä.

Tässä tutkimuksessa tutkittiin, millä mekanismeilla ohjelmistoyrityksissä tehdään päätöksiä tuotteisiin toteutettavista ominaisuuksista. Lisäksi selvitettiin haasteita, jotka liittyvät ominaisuuspäätösten tekemiseen sekä niihin käytäntöihin, joilla näitä haasteita nykyään yritetään ratkaista. Tämä tehtiin, jotta ongelma-alue ymmärrettäisiin paremmin ja voitaisiin näin luoda pohjaa käytäntöjen parantamiselle. Lisäksi evaluoitiin todellisissa tuotekehitysprojekteissa kahta kirjallisuudessa esitettyä priorisointimenetelmää, jotta menetelmien soveltuvuudesta saataisiin lisää tietoa.

Tutkimuksessa havaittiin, että käytännössä ominaisuuspäätöksiä tehdään iteratiivisesti useissa vaiheissa tuotekehitystä. Ennen tuotekehitysprojekteja priorisointia tehdään pääsääntöisesti liiketoimintanäkökulmasta muutaman tuotejulkaisun verran eteenpäin siten, että lähin tulevaisuus määritellään tarkemmin ja kauempi tulevaisuus epätarkemmin. Projektien aikana ominaisuudet priorisoidaan uudestaan mm. tuotekehityksen kyvykkyyden ja ominaisuuksien määrittelyjen kypsyyden mukaan. Käytetyt käytännöt olivat pitkälti epämuodollisia, mutta niiden keskeisenä tausta-ajatuksena oli arvioida toteuttamisesta saatavaa liiketoimintahyötyä tuotekehityksen kustannuksiin. Lisäksi pyrittiin tuotekehitysresurssien mahdollisimman tasaiseen hyödyntämiseen ajan funktiona.

Priorisointiin havaittiin liittyvän useita haasteita, joita olivat mm. ominaisuuksien liiketoimintapotentiaalin tunnistaminen, prioriteetteihin vaikuttavien tekijöiden tilannekohtaisuus sekä vaikeus saada sekundäärisiä osapuolia mukaan päätöksentekoon. Keskeisenä haasteena oli myös prioriteetti-informaation kuvaaminen sekä välittäminen eri organisaation osien välillä. Priorisointimenetelmien havaittiin tietyissä tapauksissa auttavan ominaisuuksien järjestämisessä, mutta samalla havaittiin myös että olemassa olevan ominaisuuslistan tärkeysjärjestäminen on vain osa ongelmaa.

Avainsanat

vaatimusten priorisointi, vaatimusten määrittely ja hallinta, julkaisusuunnittelu, ohjelmistotuotanto

Acknowledgements

The practical work for this thesis was conducted in the QURE and CORE projects at the Software Business and Engineering Institute (SoberIT) at the Helsinki University of Technology. The work was funded mainly by the Finnish Funding Agency for Technology (Tekes) and the Graduate School for Electronic Business and Software Industry (GEBSI). The support of both is gratefully acknowledged.

One of the most gratifying things in this research has been the opportunity to work with many talented and experienced people. All the current and past members of the QURE and CORE projects are certainly worthy of compliments for their ideas and the fruitful discussions that we have had. I would especially like to thank Sari Kujala for providing me honest, and experienced and wise pieces of advice throughout the research process.

One colleague, however, has been over the others. My chief Marjo Kauppinen has been such a supportive, inspiring and clever trailblazer during the years that I could not imagine better. My supervisor Tomi Männistö as well has provided me a lot of support by guiding my work and giving valuable comments, for which I am grateful to him

This work contains also significant influences from many inspiring people outside the projects that have shared their views and knowledge with me. I would especially like to thank Reijo Sulonen, Jarno Vähäniitty and Jyrki Kontio for the valuable discussions that we have had.

The role of the case companies and the individual practitioners within the companies cannot be emphasized enough. Without the software companies willing to improve their practices, and the practitioners within the companies sharing their experiences and spending their time with me, this thesis would be worth of nothing. My warmest thanks are dedicated to you all.

In my private life, I hope to be a brave enough friend, relative, and partner to express my gratitude somewhere else than in the Acknowledgements of a Licentiate thesis.

Espoo, 06.09.2006

Laura Lehtola

Contents

List of publications

This thesis is based on the following publications, referred to by Roman numerals:

I. Lehtola, L., Kauppinen, M., and Kujala, S. (2004) Requirements Prioritization Challenges in Practice. Proceedings of 5th International Conference on Product Focused Software Process Improvement (PROFESS 2004), April 5 - 8, Kansai Science City, Japan, pp. 497-508.

II. Lehtola, L., Kauppinen, M. and Kujala, S. (2005) Linking Business View to Requirements Engineering: Long-Term Product Planning by Roadmapping. Proceedings of 13th IEEE Requirements Engineering Conference (RE2005), August 29th - September 2, Paris, France, pp. 439-446.

III. Lehtola, L. and Kauppinen, M. (2006) Suitability of Requirements Prioritization Methods for Market-Driven Software Product Development. Software Process Improvement and Practice (SPIP), 11(1), 7-19.

The author of this thesis is the principal author of all the included publications and is responsible for writing all the text of the papers. Marjo Kauppinen participated in the research related to all of the papers. In addition, she provided valuable improvement suggestions and feedback for all three publications. Sari Kujala provided valuable feedback for Publications I and II.

1. Introduction

A software product development organization invests resources in product development and expects maximal added value from their investments (Boehm 2003). This means that providing value to different customer and end-user segments with products is a necessity for the business of product development companies. Providing value with the product requires, however, a successful selection of the requirements to be implemented in the products.

Requirements prioritization is defined as an activity during which the most important requirements for the system (or release) should be discovered (Sommerville 1996). In practice, only a limited set of requirements can be implemented in one release, but the product should, however, meet the needs of the customers and reach the markets in time (Karlsson and Ryan 1997; Siddiqi and Shekaran 1996). This means that trade-offs have to be made during the development work.

This thesis investigates how the prioritization and selection of requirements is organized during product development in software companies operating in the product business, and what the practical challenges involved are. In addition, the suitability of prioritization methods from the requirements engineering literature to solve these challenges is discussed.

This section describes the background of the research (Section 1.1), the research projects in which the research was conducted (Section 1.2), and the most important terms used (Section 1.3).

1.1 The background of the research

According to Boehm (2003), the ultimate sponsors of the project expect that the project's end result will be to 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. Prioritizing requirements is recognized as an important activity to ensure value provision in product development (Siddiqi and Shekaran 1996; Karlsson and Ryan 1997; Karlsson et al. 1998; Regnell et al. 2001).

By definition, requirements prioritization is an activity during which the most important requirements for the system (or release) should be discovered (Sommerville 1996).The origins of the importance of prioritization are in limited product development resources, since 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 and Shekaran 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, however, also recognized as a very challenging activity (e.g. Carlshamre 2002; Kalsson et al. 2004). 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 (e.g. Karlsson and Ryan 1997; Carlshamre 2002; Moisiadis 2002). In order to prioritize requirements successfully, domain knowledge and estimation skills are required (Kalrsson 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 (Carlshamre 2002). Political issues are discussed, too (Aurum and Wohlin 2003). For example, Wiegers (1999) argues that customers might not want to prioritize their requirements, because they are afraid of having just the most important ones met and developers do not want to admit that they are not able to implement all the requirements.

In companies producing packaged software, the prioritization of requirements is even more challenging than in 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 and Grube 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). This means that such market-driven companies need effective and business-driven prioritization practices if they are to survive.

Many approaches to requirements prioritization have been introduced in the literature (e.g. Karlsson and Ryan 1997; IEEE Std 830 1998; Wiegers 1999; Regnell et al. 2001). The prioritization approaches introduced in the literature vary from high-level prioritization process descriptions (e.g. Regnell et al. 2001) to detailed prioritization algorithms (e.g. Karlsson and Ryan 1997). Such different approaches work on different measurement scales, focus on different aspects, and have different levels of sophistication (Berander and Andrews 2005). However, Karlsson and Ryan (1997) argue that despite the recent rapid and welcome growth in requirements engineering (RE) research, managers still do not have simple, effective, and industrially proven techniques for prioritizing requirements. In the early nineties, Lubars et al. (1993) found 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.

Although a good deal of research has taken place within the requirements prioritization area (Ngo-The and Ruhe 2005), little evidence exists regarding which prioritization approaches are to be preferred over others in different situations and environments. Even though many authors have evaluated prioritization approaches and done comparison studies with them, the suitability of the approaches for solving practical product development challenges in prioritization has not been widely studied. It is even not clear if any of the techniques can solve the existing challenges in the area of requirements prioritization. Furthermore, no in-depth description of the current state of practice in requirements prioritization and the practical challenges involved has been presented in the literature.

The aim of this thesis is to investigate the current state of practice in the area of requirements prioritization in software companies operating in the product business and the relationship between industrial practice and requirements prioritization methods from the literature. The focus of this study is on how the prioritization and selection of requirements is organized in software product development organizations and what the practical challenges involved are. In addition, the suitability of prioritization methods for solving these challenges is investigated.

In practice, we interviewed 25 practitioners from seven software companies in order to understand the existing approaches that software companies have to making decisions concerning the requirements to be implemented in their products. In addition, we worked for three years in close co-operation with two companies so as to gain a deep understanding of their existing practices and challenges in the prioritization of requirements. Furthermore, we evaluated two requirements prioritization methods from the requirements engineering literature in development projects in order to understand how these methods suit for practical product development.

As a contribution, we provide information concerning the existing practices and challenges in the area of selecting requirements to be implemented in products. In addition, we discuss the suitability of the requirements prioritization methods introduced in the literature to solve these challenges. Furthermore, we outline the issues that affect the success of prioritization in software product development organizations.

1.2 The QURE and CORE projects

This research was conducted in the Software Business and Engineering Institute (SoberIT) at Helsinki University of Technology (HUT) as part of the QURE (Quality Through Requirements) and CORE (Competitive Advantage through Stakeholder-Driven Requirements Engineering) research projects. Both of the projects were financed by Tekes (The Finnish Technology Fund) and the companies participating in the project. The QURE project lasted from the year 1999 to the year 2002. The CORE project lasted from the year 2003 to the year 2006.

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 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.

1.3 Key terms

Requirements engineering (**RE**) is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families. (Zave 1997)

Wiegers (2003) defines a **requirement** as a property that a product must have to provide value to a stakeholder. He also defines a requirement as a statement of a customer need or objective, or of a condition or capability that a product must posses to satisfy such a need or objective.

Feature represents a logical unit of behavior 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. 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.

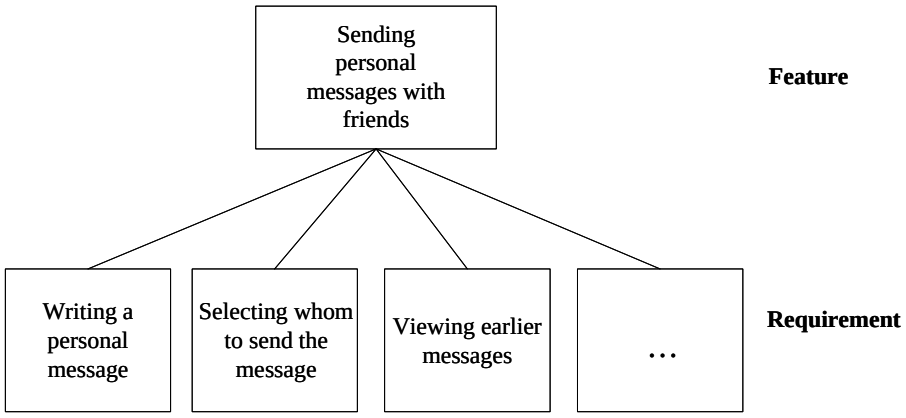

Figure 1. A feature represents a logical unit of behavior from the stakeholder's perspective

Requirements prioritization is an activity during which the most important requirements for the system (or release) should be discovered (Sommerville 1996). This concept originates from the context of the development of customer-specific systems, where all the requirements are elicited, analyzed, documented, and validated within one project. Many of the findings concerning requirements prioritization in the literature, can, however, be generalized to concern the prioritization of features as well.

To avoid unnecessary complexity, the term **requirements prioritization** is used in this thesis as a general term for both feature and requirements prioritization. Additionally, the term **prioritization practice** is used as a general term for any activity performed to find the optimum implementation order of features or requirements. For example, roadmapping and requirements prioritization are called by different names in day-to-day work. However, they both aim to find the right implementation order for the most valuable features or requirements.

Roadmapping (or product-technology roadmapping) can be defined as a 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). In practice, the links between layers are usually not explicated and the product layer of the roadmaps is mostly used. This

means, in software companies' practice, that the term "roadmapping" is most often used to describe the activities that aim to allocate the potential future features of the products to forthcoming releases. Thus, roadmapping can be seen as a product management level prioritization activity, since the aim of it is to prioritize features for projects over a period of time.

Software companies can be roughly divided in two; companies operating in the **software project business** and companies operating in the **software product business** (Nambisan 2001). Table 1 shows the terms used in different sources.

Table 1. Key terms

	Software product business	**Software project business**
Type of software	Software product Packaged software Off-the-shelf software	Tailored software Bespoke software
Type of development	Market-driven development Software product development	Bespoke software development
Type of company	Software product company	Software service company
Type of RE	Market-driven RE	RE

2. Literature study

In this section, the background to the research based on the literature is presented. Section 2.1 summarizes general issues related to software product development models, requirements engineering, and market-driven requirements engineering, and Section 2.2 summarizes issues related to the prioritization of requirements.

2.1 Software product development

In order to understand requirements prioritization in the context of software product development, the essential elements of product development models, requirements engineering, and market-driven requirements engineering need to be understood. In this section, the focus is on issues that put the prioritization and selection of requirements in context.

2.1.1 Software development models

There are two basic types of software development life-cycle models: the sequential or waterfall model (Royce 1970), and the iterative and incremental development model. The basic distinction between these two models is that in the sequential model the development of software proceeds sequentially in stages and in the iterative and incremental development model software is developed in small increments in several subsequent iterations.

Sommerville (1996) states the stages of sequential software development. These are requirements definition, system and software design, implementation and unit testing, integration and system testing, and operation and maintenance, as shown in Figure 2. According to Schach (2002), iterative software development consists of a set of iterations following each other. The phases of iteration include specification, design, implementation, and testing, which may overlap in some cases.

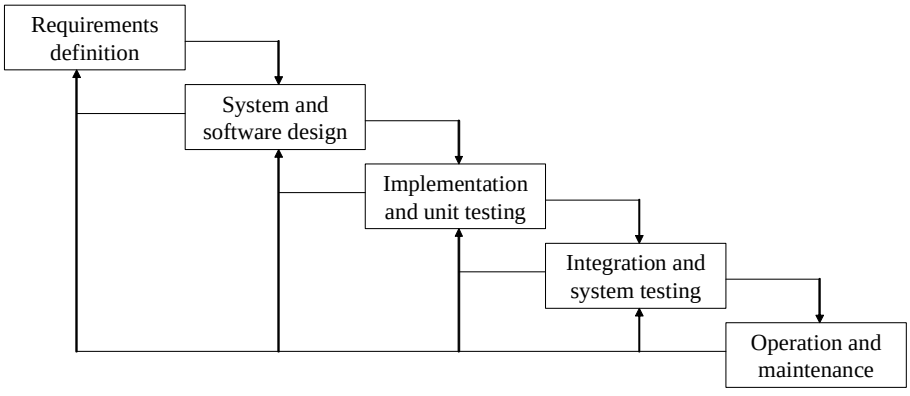

Figure 2. The sequential software development model, adapted from Sommerville (1996)[1]

[1] This diagram was drawn by Kristian Rautiainen and it was originally published in Rautiainen (2004)

The benefits and disadvantages of both models have been discussed over the years. The focus in this study is not on software development life-cycle models, and therefore we do not go deeply into that discussion. However, by understanding the basic activities common to all development models, an overall picture of software development models and software development in general can be gained. The main lesson learned here for this study is that regardless of the development life-cycle model used, a definition and selection of the requirements to be implemented is needed in one way or another.

2.1.2 Requirements engineering

Requirements engineering covers all of the activities involved in discovering, documenting, and maintaining a set of requirements for a computer-based system (Kotonya and Sommerville 1998).

Wiegers (2003) defines a **requirement** as a property that a product must have to provide value to a stakeholder. However, "requirement" is not an unambiguous term, since different authors seem to define it differently and emphasize different viewpoints in their definitions. For example, an IEEE standard defines the term (IEEE Std 610.12 1990) as:

1. A condition or capability needed by a user to solve a problem or achieve an objective.
2. 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.
3. 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 (1998) state that (system) requirements define what the system is required to do and the circumstances under which it is required to operate.

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 and Sommerville 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 (see Figure 3). 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 and Sommerville 1998). Sound requirements processes emphasize a collaborative approach to product development, involving multiple stakeholder perspectives in a partnership through a project (Wiegers, 2003).

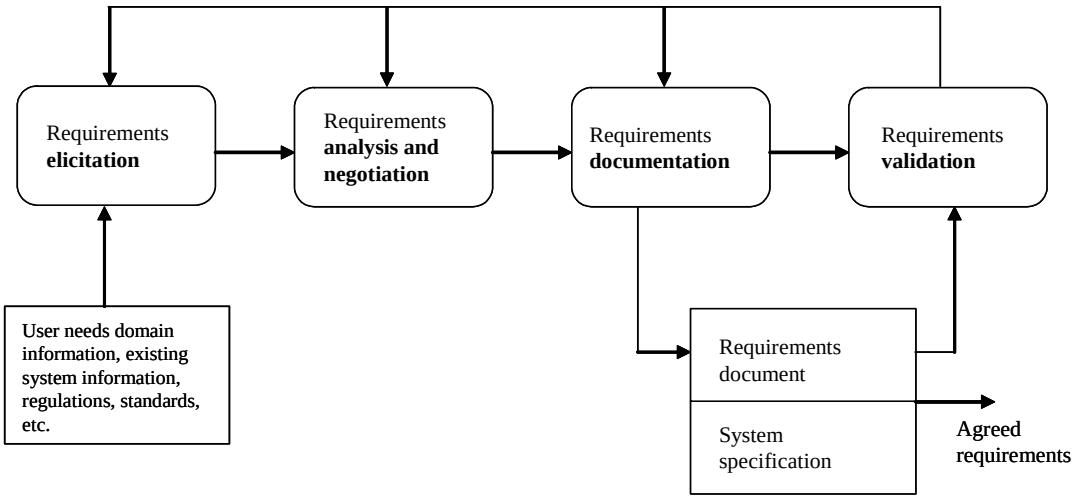

Figure 3. Coarse-grain activity model of the requirements engineering process (Kotonya and Sommerville 1998)

Requirements engineering is usually seen as the first phase of the product development cycle. For example, Jackson (1995) argues that requirements engineering and design are separate activities, since 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.

However, requirements engineering should be seen neither as one-of-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.

2.1.3 Market-driven requirements engineering

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 and Sawyer 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. What, then, are the main differences between software product and service companies? 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.

Market-driven requirements engineering refers to requirements engineering operated 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). 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.

One of the key effects that the movement to a product business has had on requirements engineering is the increasing importance of requirements prioritization. For example, Moisiadis (2002) points out that "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". Sawyer also points out that when the software is offered through recurrent releases, careful release planning and requirements prioritization are needed (Sawyer 2000). Furthermore, many of the key challenges in market-driven requirements engineering reported by Karlsson et al (2003) 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).

There are also other differences between traditional and market-driven RE. 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). 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).

2.2 Prioritization and selection of requirements

In this section activities aimed at selecting the most valuable requirements to be implemented in the product in different phases of product development are discussed.

2.2.1 Role of requirements engineering decisions in value provision

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 and Shekaran 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, in the software product business, the goal 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). According to Penny (2002) 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. Additionally, Gorchels (2000) points out that one of the key activities in managing (existing) products is to add value by improving features and/or reducing costs

2.2.2 Requirements prioritization and roadmapping

Sommerville (1996) defines requirements prioritization as an activity during which the most important requirements for the system are discovered. 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. 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 (e.g. Karlsson and Ryan 1997; Jung 1998)”.

Requirements prioritization is, however, needed not only to ignore the least important requirements, but also to improve effective project management. 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 (Wiegers 2003).

Requirements engineering plays a key role in selecting and managing requirements for a given project, but the “upstream process” before the project starts often remains vague (Ebert 2005). As a whole, requirements engineering before projects seems not to be much discussed 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.

Requirements engineering decisions, (e.g. prioritization) must be linked to the business decisions of the company. According to Ebert (2005), only by integrating upstream (e.g.

roadmap) and downstream (i.e. project) processes will projects be successful. For example, Favaro (2002) points out that, since the purpose of the requirements process is to add business value, the person in the position of managing requirements is in the position of making the most of strategic opportunities. However, describing the link for every single requirement explicitly is difficult in practice.

Usually, the company or business unit strategy is described on such a high level that the distance to single requirements documents is far too great. Describing the link to strategy for every single requirement explicitly is difficult in practice. Roadmapping is one tool that companies have used to build a linkage between business decisions and product development.

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). The output of the roadmapping process is called a roadmap. A generic roadmap is a time-based chart that typically includes both commercial and technological perspectives (Kappel 2001) (see Figure 4).

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.

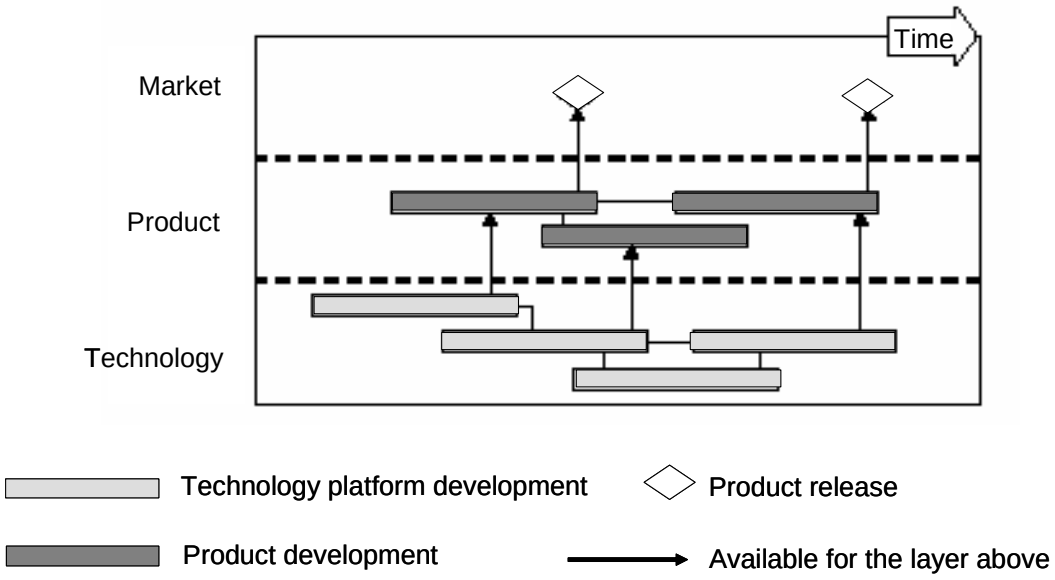

Figure 4. Roadmap provides powerful visualization of a forecast (De Gregorio, 2000)

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 and Kappel 2003). According to Phaal et al. (2002), roadmaps provide a means of charting a migration path between the current state of the business (for each layer) and the long-term vision.

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. (2002), 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. In addition, the graphical form of the roadmap is reported as being a powerful communication mechanism (Phaal et al. 2002). Compared to just textual documents, graphical roadmaps usually give a better high-level view and help in sharing the common understanding with different stakeholders.

In the case companies 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, since the aim of it is to prioritize features to releases over time.

2.2.3 Requirements prioritization methods from the RE literature

The literature offers several methods for requirements prioritization. The prioritization methods introduced in the requirements engineering literature can be roughly divided into three categories, according to the type of approach. 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 discussed.

Typically, the descriptions of the methods introduced in the literature do not specify in detail the phases of product development in which they should be used. This means that the term "requirements prioritization approach" in this context may also mean an approach to prioritizing features before projects.

Grouping methods. In grouping methods requirements are put in groups 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, called "Must", "Essential", and "Conditional".

Some authors suggest practitioners use another approach (for example 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 2. Grouping 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. Forced Priority Groups are a variation on the method mentioned above, with the distinction that only a certain percentage of requirements can be put into each group. This is done to avoid having all the requirements in the most important group.	IEEE (1998)
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 ten lists are considered the most important.	Lauesen (2002)

The grouping techniques are easy to use and learn since 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 3. 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.

Table 3. 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, where 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 and Ryan (1997)
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 and Regnell (2005)
Planning Game	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)
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 (2003)
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 (2006)

Compared to just grouping the requirements, these kinds of methods give a wider insight for prioritization and may help to take different aspects affecting priority into account better (e.g. Karlsson and Ryan 1997). 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 (for example 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 4). Voting comes to the question when there are many stakeholders that have different viewpoints on how to make this allocation (for example product managers responsible for different customer segments).

Table 4. 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.	Leffingwell and Widrig (2000)
Modified $100 test (cumulative voting)	A modification of the method described above is that stakeholders get a different amount of money from each other. For example, the total amount of "money" invested in product development could be $1000 and this amount of money could be divided between business units according to the investment levels.	Berander and Jönsson (2006)

The benefit of using voting and investing is that it 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 since they just want to invest in their specialties.

2.2.4 Challenges in requirements prioritization

The prioritization and selection of requirements to be implemented in products has been recognized as a challenging activity. Many authors have found and provided explanations and rationales for the challenges involved. Here, we focus mostly on challenges reported to concern activity referred to as requirements prioritization. However, most of these challenges can be generalized so as also to concern feature prioritization on the roadmapping level before projects.

Defining and quantifying aspects affecting priorities. A couple of authors discuss the challenge of defining and quantifying the issues that have an effect on priorities. For example, Carlshamre (2002) found that in practice decision-makers find that it is not easy to define the aspects on the basis of which prioritization decisions should be made. In addition, Bubenko (1995) points out that practitioners lack knowledge of how to quantify the benefits and risks of different alternative designs and requirements.

Making trade-offs between different types of requirements. Comparing different types of requirements with each other has been recognized as difficult. For example, Karlsson et al. (2002) identified problems in market-driven companies in having a good balance between market-driven and technology-driven requirements in their releases. However, it is necessary to achieve a good trade-off between requirements corresponding to perceived user needs and new and inventive ones that may provide a competitive advantage (Karlsson et al. 2002).

Dependencies between requirements. Requirements are very often dependent on each other. One commonly agreed challenge in prioritization is how to handle the dependencies between requirements. As Carlshamre (2002) puts it, "requirements depend on each other and priorities are always relative".

Making business-based decisions. 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. An important issue here is how to incorporate the expertise from marketing departments and the visions of top-level management in the prioritization process. In addition, Bubenko (1995) points out that links between business and enterprise models and information system specifications are usually not maintained. Furthermore, Damian and Zowghi (2003) found out that the requirements expressed by customers were often not aligned with business requirements.

Lack of effective techniques for handling large numbers of requirements. In market-driven companies, a great number of requirements are usually gathered from different sources before development projects. This causes challenges for prioritization. For example, Karlsson et al. (2002) found out that "thousands of requirements to consider resulted in difficulties when prioritizing requirements for the next release".

Lack of information. Many authors report that one key problem in prioritization is that decision-makers do not have enough information about the issues affecting their decisions (e.g. Boehm et al. 2001; Damian and Zowghi 2003). For example, Boehm et al. (2001) point out that the situation is often that requirements 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 common problem is that decision forums often do not include a representative sample of stakeholders (Damian and Zowghi 2003). However, Karlsson et al. (2002), for example, point out that improved communication between marketing and developers may improve release planning quality, which in turn increases the quality of the product.

Changes happening as a function of time. Priorities, as well as the importance of the aspects having an effect on priorities, may vary as a function of time, which causes challenges. 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).

Process issues. Many of the challenges involved in prioritization seem to be related to the process of decision-making. For example, Damian and Zowghi (2003) reported ineffective decision-making meetings as one key challenge. Process challenges also exist on the roadmapping level. Phaal et al. (2002) state a very noteworthy/important challenge by reminding us that "even though the roadmap itself is fairly simple in structure and concept, it should represent the final distilled outputs from a strategy and planning process". They also summarize the key challenges as follows: developing a robust roadmapping process, starting up the roadmapping process, and keeping the roadmap process alive.

3. Research design

This section concentrates on the methodological issues. The research questions are stated and the research approach of this study is described. Next, the data collection methods and data analysis are briefly described. Finally, the research process is presented at the end of this section.

3.1 Research problem

The research problem of this study is stated as follows:

> *What is the state of practice in the prioritization and selection of requirements in companies operating in the software product business, and can the existing challenges be tackled with requirements prioritization methods?*

The research problem is addressed through answering the following research questions:

1. What are the existing requirements prioritization practices and models used in companies operating in the software product business?

2. What are the challenges that the companies operating in the software product business face when making decisions concerning the requirements to be implemented in their products?

3. How do the requirements prioritization methods from the RE literature suit market-driven software product development?

Table 5 represents the relationship between the research questions and publications. Furthermore, two of the research questions are answered on the basis of both the study reported in the publications and extra data collected. These cases are marked in the table. The research cases are introduced in more detail in Section 3.4.

Table 5. Relationship between research questions and publications

	Publication			**Additional data collection**
	I	**II**	**III**	
1. Challenges	X			Interviews, improvement work
2. Existing practices	X	X	X	Interviews, improvement work
3. Suitability of methods		X		-

3.2 Research approach

In this section, we concentrate on the case study approach and explain the basics of the action research approach that we used in certain parts of the research. In addition, we explain the research process we used.

This research is qualitative in nature. The first part of the research is a qualitative case study that aims to understand and describe requirements prioritization practices,

problems, and needs in software product development companies. For the data collection we used qualitative semi-structured interviews in all our case companies, a focus group interview gathering participants from two of the case companies, and an action research approach type of co-operation in two case companies. The second part of the research is a qualitative analysis concerning the suitability of requirements prioritization methods from the requirements engineering literature in the product development context. We evaluated two requirements prioritization methods, each in one product development project in one case company.

3.2.1 Case study approach and action research

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). The case study inquiry relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result benefits from the prior development of theoretical propositions to guide data collection and analysis (Yin 1994).

Case studies usually combine several data collection methods. 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).

The purpose of this study is to analyze the existing practices, problems, and needs in the area of software product and release planning. The case study approach was chosen because it offers an opportunity to gain a deeper understanding of the phenomenon.

Action research can be seen as a special type of case study. In action research the client organization and the researchers collaborate in solving a practical problem while at the same time contributing to research. In action 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).

Formal research operates at a distance from the everyday lives of practitioners, 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). Action research, instead, seeks to engage “subjects” as equal and full participants in the research process (Stringer 1999).

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).

The action research approach was used as a part of both the first and second case studies (see Section 3.4.2. and Section 3.4.3.). In Case Study 1, we were developing new prioritization practices in co-operation with the case companies. In Case Study 2, we were participating in the development of the existing prioritization practices in the two companies in order to gain a better understanding concerning the needs and challenges that the companies face in that area.

3.3 The case companies

This research was performed in seven Finnish software product development organizations. Most of the case organizations involved in the study are Finnish software companies providing software products for both consumers and for professional users. The research data were collected during the years 2001-2005. The case companies and their application areas are introduced in Table 6.

Table 6. The case companies and their application areas

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

3.4 Research cases and methods

3.4.1 Overview of the research

The research consists of two research cases. The first research case consists of state-of-practice interviews in seven case companies and improvement work on existing practices in two case organizations. The second research case consists of an evaluation of two requirements prioritization methods in one of the case companies.

Not all the case organizations participated in this research in a similar way. An overview of the case studies and the participation of different organizations is given in Figure 5. The division of this research into case studies is introduced in more detail in the following sections.

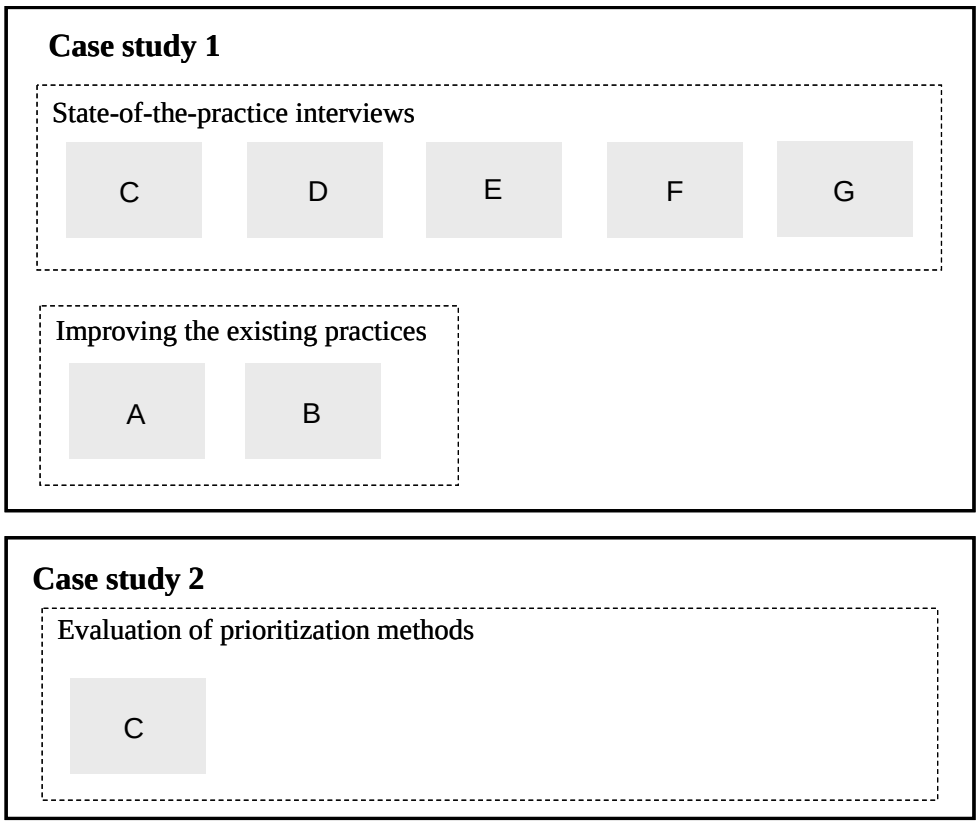

Figure 5. Overall picture of the research cases

3.4.2 Case study 1: Investigation of prioritization practices and challenges

State-of-practice interviews

The state-of-practice interviews were conducted in order 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 on. In addition, the challenges in requirements prioritization were under study. We interviewed both the main actors in requirements prioritization in the companies (here, primary stakeholders) and informants that affect or are affected by the decisions made (here, secondary stakeholders) but are not considered the most important decision-makers in the companies.

At the time of the study, the researchers co-operated actively with the case companies. Since both the researchers and practitioners had a common interest in improving existing practices, the interviewees had an interest in being honest and informative.

Primary stakeholders

Interviews with primary stakeholders were conducted in 7 companies, and in total 17 practitioners were interviewed. Nine practitioners were interviewed individually and four practitioners were interviewed as pairs. One of the interviews was a four-person focus group interview.

All of the interviews, with the exception of one of the pair interviews, were tape-recorded. The pair interview was conducted during a meeting. Because of the intimacy of the other issues discussed in the meeting, only notes were written. The interviews of which the results were not included in the publications of this thesis (I, II, or III) are marked 'x' in the "Additional interview" column.

The interviewees and their companies and roles are introduced in Table 7.

Table 7. Interviewees and their companies and viewpoints represented.

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

Secondary stakeholders

The primary goal of the interviews with the secondary stakeholders was to see how the existing challenges and practices in prioritization were seen from a distance and from different viewpoints. In addition, we wanted to get more information about the communication of priority information for different stakeholders. None of these interviews are reported in the research publications of this thesis.

The interviews with secondary stakeholders were conducted only in one case company (A). In total 7 informants were interviewed (see Table 8). Because of the limited resources we wanted to have a deep and wide understanding of the different viewpoints from one case company instead of an imperfect view from many case companies.

Table 8. Secondary interviewees and the viewpoints they represent

Title of the interviewee	**Viewpoint represented**
Business area leader	Management
Business administration manager	Management
Service manager	Service
Sales manager	Sales
Leader of product development	Product development
Marketing planner	Marketing
Team leader	Service
Documentation specialist	Documentation

Improving existing practices

In addition to the interviews, we wanted to gain a deep understanding concerning the prioritization of requirements in different phases of product development. This is why we got involved in two process improvement projects, each of them in one of our case companies (Case Companies A and B).

The aim of the process improvement projects was basically the same in both case companies. The practitioners wanted to improve their existing mechanisms for prioritizing requirements for forthcoming releases. One of the key development areas was to improve the existing practices towards being more business-driven.

At the time of the study, the authors were active participants in the RE process improvement work in the case organization. This made it possible for us to use an action research approach and, via that, to gain a deeper insight into what the practitioners do, not just what they say they do (Argysis et al. 1985). According to Avison et al. (1999), action research is an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning.

The practical role of the researchers during the improvement process was that of a kind of facilitator; practitioners asked them for advice and comments related to improving the process, as well as having spontaneous discussions about the current challenges of the organization. In addition, the researchers gave comments on early drafts of the process and templates, as well as interviewing practitioners other than the ones responsible for process development, in order to learn honest opinions about the processes. In the other case company, another researcher also participated in the data collection.

In addition to the interviews, the researchers analyzed the documents that practitioners associated with requirements prioritization in the case companies. In addition, the researcher kept a research diary during the research process in which she kept field notes along the way. The field notes can be seen both as notes from meetings as well as a type of preliminary analysis.

3.4.3 Case study 2: Evaluation of the requirements prioritization methods

In order to improve the prioritization practices in Case Organization C, 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, evaluate it, and 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.

We evaluated two existing requirements prioritization methods, the pair-wise comparison technique (Karlsson 1996) and Wiegers' method (Wiegers 1999), in real development projects. Both of the studies were performed in separate projects that posed different challenges. 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.

In both cases, we selected relevant participants within or outside the project and a subset of the project's requirements. 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 practical role of the researcher in the improvement process was to function as a kind of a facilitator or moderator. She introduced the prioritization methods to the improvement group, gathered and documented the experience, presented the findings from the cases to the group members, and supported the improvement work. In addition, she interviewed the usability expert before the improvement group was established, in order to have a wider understanding of the situation in the organization at that time. She also interviewed the practitioners about the usage of the method and how they felt about the prioritization results that the methods gave.

Evaluation of Pair-wise comparison

Project Alpha was an investigation project during which the organization wanted to have more information about real user preferences concerning one of the main products of the company. At the time of the study, the practitioners involved had gathered a good deal of data concerning user needs by performing a field study (introduced in Kujala and Mäntylä 2000) and they were satisfied with their manifold new findings. However, all the user needs had an equal value in the report and the practitioners did not know how to select the most important needs for further recognition. They wanted to introduce a common prioritization method to the project in order to systematically prioritize the user needs.

Four users of the product were invited to prioritize the gathered user needs by means of the pair-wise comparison technique (see Karlsson 1996), while the project manager wanted to know which of the requirements are the most important to the users. We grouped the requirements into ten categories of 20 or less requirements. The categories were such as "Usability" or "Performance". We predetermined all of the users to prioritize the same four categories of requirements. However, we found with the first user that it was impossible to prioritize the biggest category. That is why we left the biggest category out with the next three users, but brought one new category in. At the end of the actual prioritization work, the practitioners were asked to complete a questionnaire concerning their experiences and feelings towards the prioritization method they used.

Evaluation of Wiegers' method

In Project Beta, there were numerous change requests that had been gathered by a project manager, but no ideas as to how to systematically prioritize these. The project manager had tried to prioritize the requirements on the basis of his experience, but felt that he needed a systematic way to find which of the requests to implement first.

In this case, both the project manager and another project member prioritized 6 change requests using Wiegers' method (see Wiegers 1999). In this case, the practitioners wanted to use the method as described in the literature. However, as they faced some challenges concerning the usage of the method during the prioritization, the product manager made his own adjustment to it. This is discussed in more detail in the Results section.

3.5 Data analysis

In Case Study 1, all of the interviews except one of the pair interviews were transcribed on the basis of the recordings. Soon after the interviews, the most significant findings from each interview were moved to mind maps for further processing. The mind maps were completed on the basis of the transcriptions and the opinions of the interviewees were compared to each other in order to find similarities and discrepancies. In addition to comparing all 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 companies, the researcher kept a research diary in which she kept her field notes. Case write-ups were written after every meeting. The field notes were mostly descriptions about half a page long of what happened in each meeting and, typically, a few sentences of analysis of observations concerning, for example, conflicts between participants or the motivation of participants. The field notes can be seen both as notes from the meetings and as a type of preliminary analysis.

In Case Study 2, the practical role of the researcher was that of a facilitator. Therefore 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 these informally with practitioners after the prioritization sessions. The questionnaire also provided important additional data about practical experiences and attitudes towards the prioritization methods. The findings from the cases were compared to each other in order to find similarities between them.

4. Results: State of practice in prioritization

In this section, the current state of practice in the selection and prioritization of requirements in the software product companies is summarized.

4.1 Requirements are prioritized in many phases

In the case companies of this study, requirements were prioritized in many phases of the development work. These phases can roughly be divided into two:

- Product management-level practices
- Project-level practices

Product management-level practices are prioritization activities that are performed before development projects. This phase was commonly called *roadmapping* in the case organizations. Project-level practices are prioritization activities that are performed within projects. Figure 6 illustrates this division.

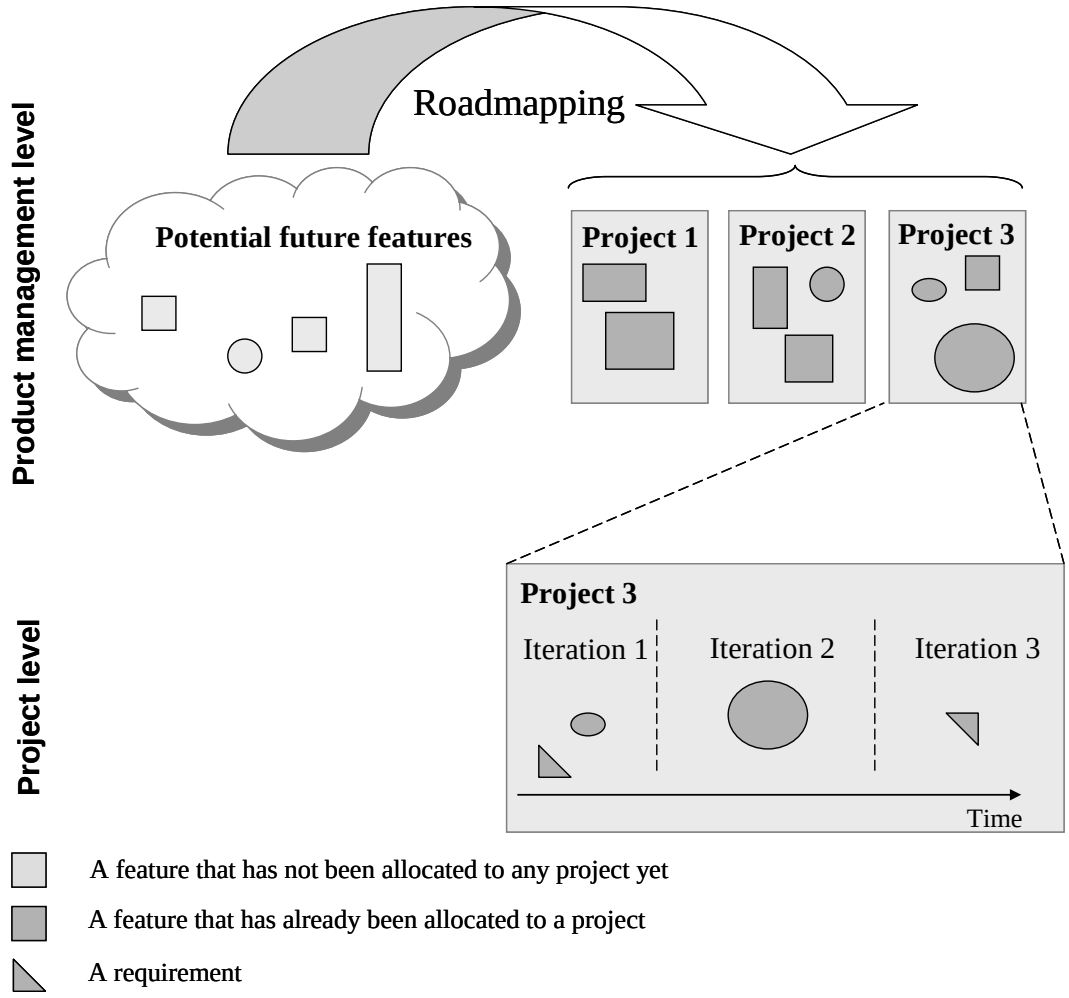

Figure 6. Requirements are prioritized in many phases

Typically, on the product management level the goal of requirements prioritization 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. Almost all of the case companies interviewed received a great number of ideas for potential features from the markets that were analyzed and prioritized to certain product releases (typically called projects). In most of the case companies these ideas were collected into an RE tool by different internal stakeholders. In this phase, product managers were typically the main actors trying to allocate the manifold requirements gathered from the field for forthcoming releases of the product.

On the project level, within one project, the main goal of the prioritization was to organize requirements that had already been selected to be implemented 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 were most often project managers who, in co-operation with product managers and software architects, allocate requirements to be implemented in the next iterations of an ongoing project.

4.2 Needs for product management-level prioritization

In the literature, requirements prioritization is typically introduced as an activity performed within a software development project[2]. Requirements prioritization is usually seen as an activity that is needed to put requirements that have already been selected in order according to their importance or urgency. In this section, we describe why the case companies of this study also needed prioritization on the product management level before the development projects.

Widening the focus from short-term plans to more long-term thinking. The starting point for the development of the roadmapping processes in the two case organizations was that practitioners needed a way to make more strategically oriented decisions about the future features of their products. Charting a migration path between the current state of the business and the long-term vision was seen as important. The practitioners felt that they would need some articulated decisions, not only concerning the future features but also as to what the customer segments are, the geographical areas that they want to satisfy most. This was needed so as to be able to prepare for the future and to understand the priorities better. Figure 7 illustrates the purpose of long-term product planning between business information and requirements engineering.

[2] This is mainly because of the historical rationales introduced, for example, in Ebert (2005)

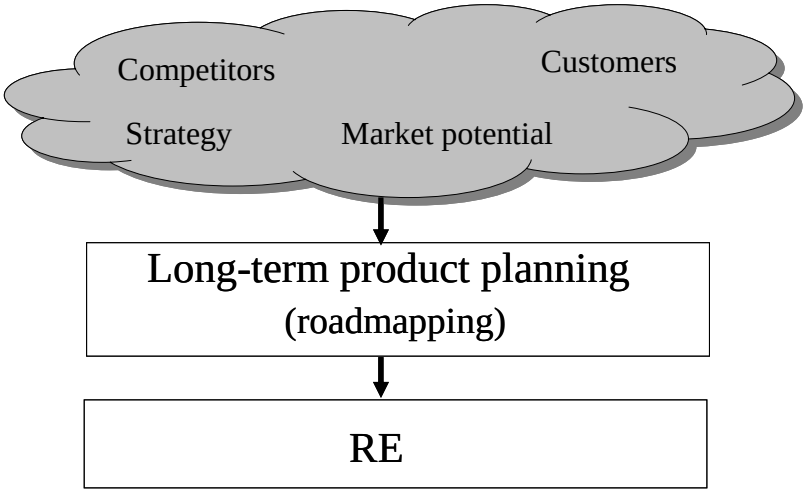

Figure 7. Linking business view to RE via long-term product planning

Gaining and sharing a holistic long-term view of the future with different stakeholders. Sharing a common and holistic long-term view of the future development steps of the software products within the organization was recognized as important in the case companies. All the case companies 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 products' future development steps and their rationales were critical issues that the companies had had problems with.

One important need for many of the case companies was that they wanted to inform and involve more stakeholders (R&D, marketing and sales, and management) within the company earlier in the development cycle. The companies wanted to start the development by setting the high-level targets beforehand and co-operatively so that the marketing and sales functions were able to prepare their activities at the same time as the product development activities were taking place.

Explicating the links between the development needs of different products. In most of the case companies, the product managers were fighting for the same product development resources. This means that the practitioners needed to be able to communicate their future ideas and resource needs to other product managers in a way that is understandable to the others. This communication was also needed to find and realize potential synergies that may exist between different products and their future development directions.

In addition, the management in the case companies faced the challenge of comparing different projects and feature ideas with each other. This is why, in many of the case companies, the management had compelled practitioners to prepare the roadmaps in such a way that the management can see all the planned releases in the same format and thus be able to compare them with each other in order to make trade-offs.

4.3 Characteristics of prioritization practices

In this section, the characteristics of the prioritization practices are discussed. Section 4.3.1 summarizes the characteristics of practices on the product management level and Section 4.3.2 on the project level.

4.3.1 Characteristics of product management-level prioritization practices

Business viewpoint dominates the decisions. At the product management level the prioritization was mostly performed from the business viewpoint in the case companies. This means that the most important aspects affecting requirements priorities considered at this phase were business issues, such as the value that implementing a requirement could provide to the customers and thus to the company. In a few special cases, the available resources were so limited and specific that the selection of the requirements to be implemented was made more from the viewpoint of available resources than from the business viewpoint.

Typical planning horizon is few releases ahead. The most typical planning horizon in the case companies 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 more detail. The remote future was also outlined, but in less detail. The actual practices used to perform this kind of planning varied between the case companies. (Examples of these are introduced in Section 4.4.1)

Focus is usually on features of one product. The main content of roadmaps in the case companies was most often the forthcoming features of products on a high level. In a few case companies, the roadmap templates also covered issues such as 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.

Most case companies saw, for example, language versions and dropped features (compared to earlier versions) as important information that should be covered in the roadmaps. However, integrating these issues into roadmaps was not yet that systematic in the case companies. In addition, the roadmapping was usually done from the viewpoint of one product and the linking between products was not explicit.

Preparation of roadmaps is mostly product managers' responsibility. The persons responsible for preparing the roadmaps were usually product managers, which is quite natural since the management of the products' future was their responsibility. The most important stakeholder groups to which the contents of roadmaps should be communicated and with which the product managers should negotiate were usually seen as management, sales and channel partners, and customers. However, the viewpoint of the developer was usually less emphasized in the case companies than that of other stakeholders.

4.3.2 Characteristics of project-level prioritization practices

Many iterative prioritizations during the project. The starting point for the first project-level prioritization in the case projects had usually been the priorities nominated from the business viewpoint by product managers. However, as the project 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. The aspect that usually dominated the decisions within the projects was how mature the definitions of different requirements at different points of the project were.

Project managers and software architects are main actors. The project managers and software architects seemed to be the main actors on the project level. The software architects usually had the best technical knowledge and understanding concerning the development costs of individual requirements. The project managers, on the other hand, were responsible for staying within the project budget frame and reaching the project business goals.

4.4 Example prioritization practices from the case companies

This section introduces the prioritization practices that were found in the case companies of this study. Section 4.4.1 introduces the product management level found and in Section 4.4.2, project-level practices are discussed. In Section 4.4.3, the mental models that decision-makers on both levels used as bases for their prioritization decisions are introduced.

4.4.1 Product management-level practices

Priority lists of local offices. The case companies that produce software products for mass markets have geographically distanced local offices that are responsible for selling the software in their market. The local offices are interested in influencing the future development steps of the products, since their existing customers demand new features and they might see potential for increasing their sales via new functionalities in their products.

Two of the case companies in this study had a practice whereby the local area offices provide the headquarters with a list of the most important requirements for their customers according to their understanding. In another case company, the local areas gave their requirements and numbers indicating their importance from 0 to 100. In another case company, the responsible person from the local area office loaded the requirements into the requirements management system and arranged them in priority groups from the local area viewpoint.

Rationales written by local offices. One of the case companies required their local offices to write rationales that indicate the business impact of the requirements that they suggest. The company had defined three fields that had to be completed for each requirement in their requirements management system. Two of them (“Business impact for the company” and “Business impact for the customer”) were completed by the local office person suggesting the requirement to the headquarters. The third field (“Cost and time estimation”) was completed by product managers or sometimes by the technical experts. This practice had improved the quality of the communication between local offices and product management at the headquarters. In addition, it had improved the transparency of the status of requirements on the way through projects. “Local offices put an effort into writing the business rationales, since they know that these affect their chances of getting their requirements through to projects. It is also easier for us (project

managers) to compare different ideas from the markets when we have these written business rationales."

Open-ended multi-release planning. Three of the case companies, which produced commercial software for mass markets, used a kind of "open-ended planning" technique for decision-making concerning the future development steps of their products. This means basically that the time horizon for future planning was open in such a way that the steps for the nearer future were planned in more detail. The remote future was also already outlined, but in less detail. The actual practices for this kind of planning varied between the case companies. The next three examples illustrate the practices that were used in three case companies.

Case Company E: "Three project baskets"

Twice a year, the product managers lead the planning of the next three product projects at feature level. Each time, the following three projects are planned. This means that each of the forthcoming releases is planned three times in total. Each iterative planning round reaches a more detailed focus for the project. The product managers validate their ideas for new features with representatives of different customers. "Our roadmap is at least semi-public and we discuss it in an ongoing way with our customers. It is possible to modify projects according to the feedback", explained the product management director.

Case Company A: "Technical roadmap"

In Case Company A, an open-ended resource and release allocation for future features was an ongoing practice. A document called a technical roadmap consisted of six big Excel spreadsheets, one for each technical aspect of the product. Within each sheet the individual product part was divided into about 5-6 sections, and each section consisted of a set of low-level requirements. Above each sheet there was a timeline divided into quarters of the year. Product managers used this roadmap so that the next release was always in focus, but they sometimes allocated features and resources roughly for subsequent releases as well.

Case Company D: "Updating requirements' status in a tool"

In Case Company D, each requirement in the requirements management system was given a status by product managers. Three different types of status for the requirements can be given. The first of these is "the exact version of the product where the requirement will be implemented". The second one is "not yet", which means that the requirement is not totally ignored but has to wait. The last section is "Never", which means that the requirement does not fit into the product strategy and will not be implemented.

4.4.2 Project-level practices

Grouping requirements according to their importance or urgency. The requirements of one project were usually nominated in three or four priority groups (for example: high, medium, low) at the beginning of the project in most of the case companies. This was done in order to communicate priority information on the requirements to developers and to make sure that the most important requirements are implemented first. "We divide the projects by milestones. The project manager is responsible for making sure that the requirements nominated as the most important ones are implemented before the first milestones," explained one of the interviewed product managers.

Negotiation in project meetings. Negotiation in project meetings was used especially in those situations where the project group was not able to implement all the requirements in the given time. These discussions were informal. The project group just made the decision as to whether there was or was not time to implement a requirement in this particular release. “We have a person who knows what it takes in the way of resources to implement the requirement and a person who knows how much effect it has on business. It is just a mutual discussion,” explained one project manager. In some of the case companies, the product manager and the steering group were also involved in these decisions.

Impact validation. In one of the case companies, proposed features and changes were evaluated by an Impact Estimation method modified from the one introduced by Tom Gilb (www.gilb.com). This practice was used in order to validate the extent to which the features contribute to the purpose of the project, and to compare alternative features with each other. In practice, this meant that new feature ideas or change requests were validated according to certain criteria defined for the project. For example, in one project, the requirements were evaluated according to a total of 17 criteria, such as “time-to-market” and “reviewed by customers”.

Implementing the most mature requirements first. In many of the case companies, not all of the requirements for the project were defined when the projects started. Practitioners in one case company explicitly mentioned that this caused situations where the most mature requirements from the definition viewpoint were implemented first, no matter what their priorities according to business impact were. This was because the immature requirements were impossible to implement before a better definition of them had been obtained. In the same case company, requirements were usually allocated to developers in weekly meetings. This means that the skill profile of the developers also had an effect on the implementation order of the requirements.

4.4.3 Mental models used on both project management and project levels

In this section, the mental models that practitioners used both before and within projects as a basis for the selection and prioritization of requirements are discussed. Here, the term “mental model” means a model that practitioners more or less implicitly use as a basis for their analysis of the priority of the requirements.

Mutual cost-value analysis. Many of the case companies 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.

Modified Kano model-based analysis. One of the case companies 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 do not have that much 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 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.

4.5 Practical challenges in prioritization

In this section, the typical challenges that companies face in the prioritization and selection of requirements are introduced. The challenges are organized into three groups according to their type. First, challenges related to combining information are introduced. Second, people- or resource-related challenges are introduced. And third, communication- and description-related challenges are discussed.

4.5.1 Information combining-related challenges

Priority being a combination of many aspects affecting it. The requirement’s or feature’s importance to a customer was an important, but usually not the only, aspect that had an effect on its priority in the case companies. The case organizations faced challenges in defining which aspects should be taken into account when setting priorities.

Our study indicates that there are three main points of view affecting priorities which are more or less explicitly taken into account when setting priorities. These three points of view, introduced in Figure 8, encapsulate the other aspects found during the research.

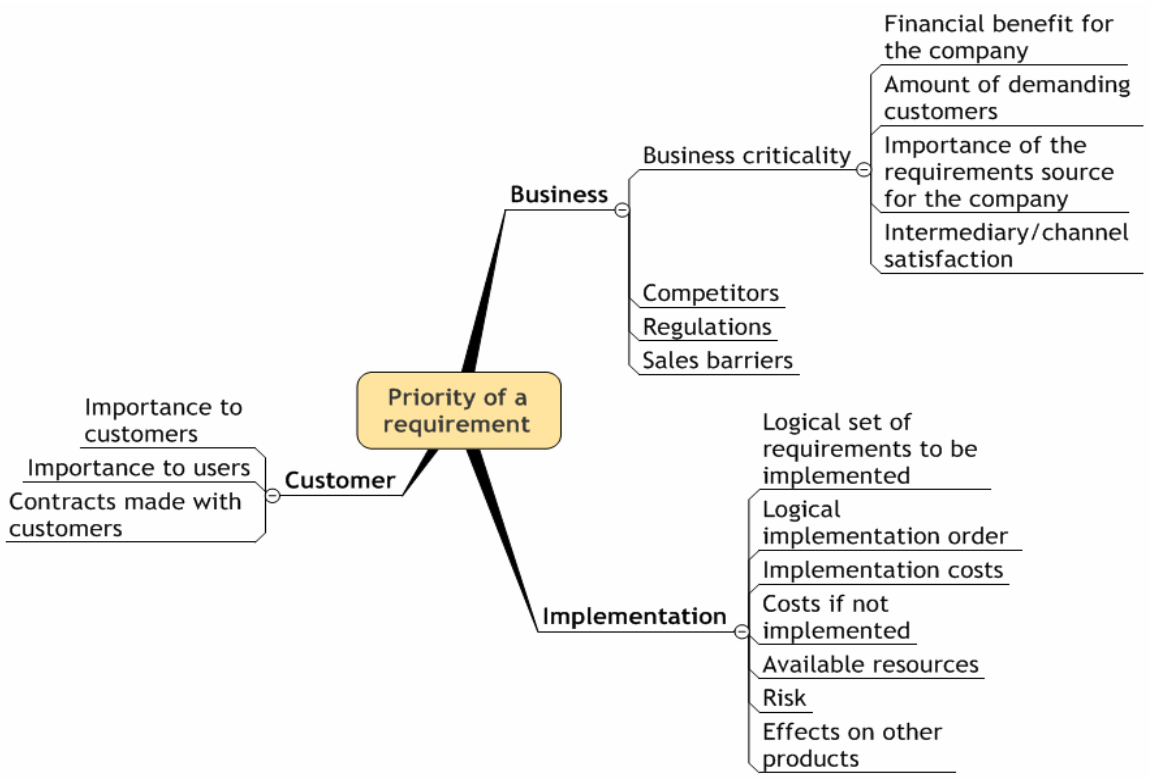

Figure 8. Aspects affecting priorities

For a company it is a lifeline to profit, so issues such as customer relationships, competitors, and the importance of the requirement's source for the company were mentioned as issues that have to be taken into account. Another point of view is that of customers and users. The development organization must know which of the requirements are most important to them. A third point of view is implementation. The resources of the company and the manufacturing situation, as well as the logical implementation order, were seen as having an effect on which requirements should be implemented first.

The number of aspects affecting requirements priorities seems to expand in market-driven companies, compared to those companies operating in the 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 the project business. In addition, it is not always easy for practitioners to decide which aspects and stakeholders should be taken into account when setting priorities, getting the right information about them, and combining information from different sources.

4.5.2 People- or resource-related challenges

Involving different stakeholders and combining their knowledge. Combining knowledge and information from different stakeholders is important in the prioritization of requirements. However, it is not always easy. In the case organizations, sometimes important stakeholders such as people in the sales organization found, for example, long-term planning to be something that is not that important for them, because of the more urgent need to conclude their sales cases. Even when the stakeholders were willing to

participate in planning, it turned out to be a challenging task. Practitioners from isolated parts of the organization had different opinions and preferences, since they were responsible for different types of activities and have different planning horizons.

However, since a single person may have a limited view and insufficient access to relevant information, it is important to combine the different types of expertise that the practitioners have. For example, product developers typically have a thorough understanding of the costs of the implementation of potential future features, but not such a deep understanding of the business environment of the product. Business intelligence people, on the other hand, usually have deep market knowledge, but no clear understanding of the technical possibilities or development costs of their ideas.

Requirements prioritization being an ambiguous concept. Although it is essential that people have a common understanding of the terms they use and the activities they perform in product development, the terms “requirements prioritization” and “priority” have several different meanings in practice. This causes confusion and misunderstandings among product development personnel. The terms are not uniformly defined in organizations, so in spoken language different activities with different purposes are referred to by the same terms. This happens without the practitioners being aware of it.

The activity called “requirements prioritization” had many meanings in the case companies. Occasionally, the term was used with the meaning “How do we decide which requirements are the most important ones for the company in the long run?”; sometimes it meant “How do we decide which requirements we have to implement right away in the next product release?” or “How do we select the requirements that will be implemented first in this project?”, or “Which of the requirements describe the system in high-level terms?”.

In the case companies, 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. In some cases these two scales, the importance scale and the time scale, were equal to each other. However, in release planning other things than importance to the customer, for example the interdependencies of the requirements, can have a greater effect on implementation decisions and their schedule.

There were ambiguities not only with the usage of terms, but also with the usage of prioritization scales. The categories high, medium, and low were experienced as being ambiguous. For example, one of the interviewees mentioned that “We needed a lot of discussion about the meanings of each priority level with the project group when we set priorities. I was surprised how different the meanings we had in our minds were.”

Not enough resources to analyze every single requirement in detail. The practitioners mentioned that in practice there is no time to figure out all the relevant information as a basis for priority decisions. The development personnel had, for example, difficulties in analyzing all the raw requirements they gathered from customers. “There is no time to analyze thousands of wishes. Much of the work is done intuitively,” said one of the interviewees. In most cases the requirements specification is written by only one person. That leads to situations in which the writer of the specification thinks that there is no need to prioritize the requirements any more. One of the interviewees complained that “The writer does not want to drop anything. The first version (of the requirements document) is, in a way, just the high-priority requirements according to his or her understanding.”

4.5.3 Communication- or description-related challenges

Developers do not know enough about customer preferences. The product development personnel would like to know why a requirement is important to users or customers. Usually they have no idea, because people are working separately in product development; product development personnel do not have direct contacts with users and customers. In addition to this, there are no common practices to communicate customer and user information through the product development process.

Particularly in small projects, contact with customers and users was felt to be too narrow. A great deal of important information was gathered from users by means of the help desk calls they made. In addition to this, product development staff communicated with vendors and gathered information in this manner. In the case companies, product managers created the first requirements specification on the grounds of discussions they had had with customers. They had an idea which requirements were important to customers and placed requirements into priority categories.

There were no generally agreed 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 the project manager and the 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's point of view does the prioritization."

Linking different levels of roadmapping / prioritization. Almost all of the practitioners interviewed felt that feature-level roadmapping is not enough, and that a more rational business-oriented view of the future would be needed in their company. However, the links between business rationales and prioritization decisions seemed to be quite difficult to define. Explicit linking between different levels is needed, for example, to get an understanding of which business targets will be affected if some features are not implemented. In addition, the practitioners found it important to know explicitly which products are affected by different business targets. This complex network of responsibilities between the business and technical viewpoints is shown in Figure 9.

Balancing the available product development resources between different products was experienced as challenging in many case companies. Since product managers fight for the same resources, they needed to give early resource allocations and provide business rationales for the features they propose for implementation. However, resource estimation seemed to be difficult in the early phases.

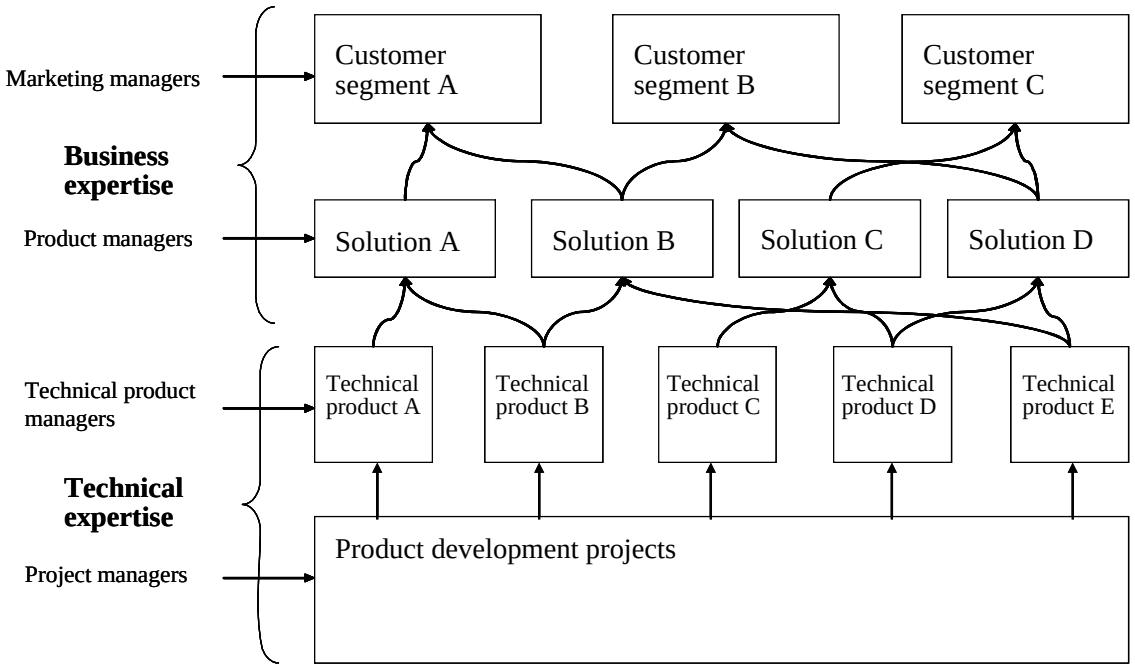

Figure 9. Complex network of responsibilities between business and technical viewpoints.

Describing the planned future development from the customer's viewpoint. Most of the case companies found it challenging to communicate the future development steps of their products from the customer's viewpoint. This kind of view was needed in order to give both the sales organization and existing and potential customers information about the future features of the products and the benefits that the customers will get. However, formulating the value provided for customers was difficult. In addition, finding the right format for visualizing and communicating this information is a challenge. Furthermore, the relations and role differences between internal roadmaps and external roadmaps are not defined in many case organizations.

4.6 Improvement ideas for tackling challenges in prioritization

In this section, improvement ideas for tackling some of the existing challenges in the prioritization area are introduced. Only those improvement ideas that have been used at least in one of the case companies are introduced here.

Market segment-based planning. One of the case organizations changed their organization during the study from a functional organization to a more customer segment-based organization. This means that after the change every customer segment has its own segment team, which includes representatives from many internal functions (product management, sales, marketing, and service). The team is co-operatively responsible for making segment development plans, including forthcoming product development plans. The practitioners felt that this type of organization had brought experts from different functions nearer to each other and improved communication. In addition, the shared responsibility forced the practitioners truly to co-operate with each other.

Including other issues than software features in roadmaps. Some case companies found that it is a good idea also to document features other than software features in

roadmaps. For example, if a company is entering new markets and needs to launch a new language version of their product, the people responsible for making the language versions have to get this information as early as possible. In addition, when they see all the resource needs, for example, for the documentation team at the same time, the prioritization of their activities is easier to organize. On the other hand, some of the future features are such that there is no point in implementing them if there are no resources to have marketing materials or user documentation concerning the issue ready.

Explicit links between different levels of roadmapping. One of the case companies built explicit linking between business and product decisions. Each product development item in the release roadmap needed to have an explicit link to one or many business-level decisions, which were also documented. Without a link to a business target a feature idea was not accepted. The practitioners in the company felt that this had increased their ability to make real business-based product development decisions.

Feedback sessions with customers. A couple of the case companies realized that it is a good idea to gather feedback from customers concerning the contents of future releases during site visits. In these companies, it was typically the product managers who visited customers with the early roadmaps for the next few releases. These companies also found that it is possible to gather totally new feature ideas during the same site visits. However, they found that it is important to explicitly emphasize to the customers that the plans introduced may change.

4.7 Summary of the state of practice

In this section, a summary of the state of practice in the area of requirements prioritization in the case companies is presented.

Prioritization practices in the case companies are mostly informal and dependent on individuals. No explicit methods for taking different aspects into account in prioritization are in use in the case companies. Not even the aspects one should take into account when deciding priorities are commonly explicated. Instead, the development personnel try to make rough guesses as to which requirements are the most important ones to customers and users, how profitable the requirements are to their own company, and how all this coheres with the strategy of the company, but there are no systematic practices for these analyses. Contracts made with customers and promises given to them in informal discussions play a major role when priorities are being set. Roughly speaking, individuals make prioritization decisions mostly on the basis of their tacit knowledge or feelings both on the product management level and project level.

Requirements prioritization seems not to be a one-off activity in practice. Instead, requirements are prioritized in many phases during product development. These phases can roughly be divided into two, the product management level and the project level. In addition to the traditional within-project prioritization, prioritization activities are also needed before projects in order to widen the focus from short-term plans to more long-term thinking, to share a holistic long-term view of the future with different stakeholders, and to explicate the links between the development needs of different products.

The practices on the *product management level* most often seem to aim at selecting the most valuable requirements from a huge amount of raw requirements. The requirements are usually allocated to a few forthcoming releases so that the time horizon for future

planning is open. The practitioners try to make this selection by getting as much information as possible from the end users and customers and understanding their needs. On the *project level*, the goal of the prioritization is to find a feasible implementation order for the requirements based on e.g. the available skills and maturity of the requirements. The practices used are mainly informal, but most often the basis of them is to evaluate the potential business value of requirements versus their implementation costs. In addition, companies aim to utilize their product development resources evenly as a function of time.

Prioritization involves manifold practical challenges. The biggest challenges on the product management level are usually to get a clear understanding of how important different requirements are for different customer segments and why. In addition, balancing the available product development resources between different products and between different customer segments, as well as forming feasible sets of requirements from the technological viewpoint to be implemented in the following releases, is essential, but difficult. The biggest challenges in requirements prioritization within projects are usually due to a lack of information concerning the business rationales of the priorities given by business stakeholders. In addition, a lack of detail in requirements definitions sometimes makes prioritization difficult, since it is challenging to estimate, for example, implementation costs without knowing in detail what is to be done.

The case companies in the study have implemented some improvements in order to tackle their current challenges. These include moving towards market segment-based planning from just product-based planning, explicit linking between different levels of roadmapping, and site visits to customers' sites. In addition, moving from software-feature roadmapping towards a whole-product view has been recognized as beneficial.

5. Results: Evaluation of the prioritization methods

The next two sections summarize the lessons learned from the case in which two requirements prioritization methods were evaluated in development projects. In Section 5.1, we discuss the benefits that practitioners felt that they got by using the prioritization methods. In Section 5.2, we point out the challenges involved within the cases in order to shed light on the limitations of using prioritization methods in product development projects. The detailed results of both cases are introduced in Publication III.

5.1 Benefits experienced by practitioners

Getting a more controlled way to evaluate the requirements. The practitioners found the idea of evaluating the requirements in a controlled way from different viewpoints interesting. Especially in the Wiegers' case (Project Beta), where each of the requirements was evaluated according to its 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. In the pair-wise comparison case (Project Alpha), practitioners were happy to see the differences between the opinions of different users.

The idea of having requirements distributed into a descending order. 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 into a descending continuum as an outcome. In the pair-wise comparison case, the product developers found the idea of getting a list of requirements, put in descending order according to their importance to users, attractive. In the Wiegers' method case, too, the practitioners were interested in having every single change request distinct from the others according to its priority. However, combining the different stakeholder preferences from the priority lists was found to be difficult.

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 expressed their preferences and opinions about the products freely during the work.

5.2 Challenges experienced by practitioners

Limited scalability of the methods. It seems that the scalability of the evaluated requirements prioritization methods is quite limited. In addition, their suitability for different types of situations seems to be quite low. In our cases, the scalability of the pair-wise comparison technique in terms of the amount of the requirements in particular was found to be quite low (< 20). Furthermore, the requirements of different levels of abstraction caused trouble. Detailed descriptions of the problems that occurred are listed in bullet points below:

Mistrust of results. 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.

The mistrust of the methods was expressed differently in the two cases. In the Wiegers' method case, practitioners changed their estimates in order to get a "better" priority order if the results given by the method seemed wrong. We found that the practitioners had strong opinions about the priorities and that in some cases they tried to use the prioritization method in order to prove their existing views. For example, if a requirement they preferred got a low priority according to the formulae, they changed their estimates so that the preferred requirement became more important in the priority list.

In the pair-wise comparison case, the users argued that pair-wise comparisons are gratuitous and that they could just select the most important requirements without comparisons. Two of the users felt that it would have been easier for them just to select the most important requirements or put the requirements in descending order without any pair-wise comparisons.

Lack of clarity concerning the usage of the methods affecting the results. Aspects of the usage of the prioritization methods that are unclear 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.

The prioritization results are never better than the raw data inserted. If a user evaluates the factors in a hurry without careful consideration or cannot nominate the extent to which one requirement is more important than another, or if a practitioner does not know what a requirement's real value for the customer is, the prioritization results are nothing more than just rough guesses. Getting exact numbers or fractals as an outcome does not ensure the validity of the results, since the mathematical calculations cannot improve the quality of the raw data inserted.

Results leading to a wrong impression of what to do on that basis. Requirements prioritization methods give a priority list as an outcome. These lists may, in some cases, lead to a false impression among practitioners as to what should be done on the basis of them. 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.

6. 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 are the existing requirements prioritization practices and models used in the companies operating in the software product business?
2. What are the challenges that the companies operating in the software product business face when making decisions concerning the requirements to be implemented in their products?
3. How do the requirements prioritization methods from the RE literature suit market-driven software product development?

6.1 Existing prioritization practices

Our study indicates that instead of being just a one-off activity, requirements prioritization is needed in many phases of development work. On the product management level, before development projects, requirements are usually prioritized from the business viewpoint. The goal of this prioritization is to allocate the most valuable requirements from business viewpoint to be implemented in the few following releases. In an individual development project, requirements are usually prioritized again from the implementation viewpoint in order to reach the best possible fit between resources and needs. Additionally, Boehm et al. (1998), for example, suggest an iterative approach to prioritization. However, not much attention has been paid in the literature to product management-level prioritization. Prioritization has been seen more as a one-off activity performed in the requirements analysis phase within a software project (Sommerville 1996).

According to our study, prioritization practices in the companies seem mostly to be informal and dependent on individuals. This is true on both the product management level and project level. Similarly, Karlsson and Ryan (1997) state that managers still do not have simple, effective, and industrially proven techniques for prioritizing requirements. Lubars et al. (1993), too, found in their study that companies do not know how to set and modify priorities. Although no explicit requirements prioritization methods were in use in the case companies, many of the practices that were found aimed at performing a rough cost-value analysis for the requirements. Some of the practices found in the case companies also addressed matters other than decision-making challenges in prioritization, such as the communication of priority information through the organization or involving stakeholders in the decision-making process.

Interestingly, it also seems that, in practice, requirements prioritization rarely involves comparing existing sets of requirements on the same level of abstraction with each other. In practice the pool of requirements is not often so strictly defined and clear. In practice, the requirements management systems in software companies are overloaded by requirements from different markets and local offices. The question that usually arises is: ‘Is it cleverer to implement this one big requirement or these eight small requirements that do not get very high scores alone?’

Furthermore, it seems that one important issue in requirements prioritization in practice is the communication of priority rationales through the organization. Product managers need to know why the local offices consider some requirements more important than others in order to make their decisions concerning the future development steps of the products on the basis of different views. In addition, project managers want to see the big picture of the priorities when planning the milestones of projects.

6.2 Challenges in prioritization

Our results confirm that requirements prioritization is a very challenging activity. Our findings support earlier studies that described requirements prioritization as "one of the most difficult tasks facing the practitioners" (Yeh 1992), including "a great deal of invisible decision-making" (Aurum and Wohlin 2003) and "politics" (Andriole 1998), and requiring "complex decision-making" (Karlsson and Ryan 1997; Carlshamre 2002).

Clemen (1996) defines four ways in which a decision can be hard. These ways are 1) the complexity of the problem; 2) uncertainty inherent in the situation; 3) the decision-maker being interested in working towards multiple objectives but progress in one direction impeding progress in others, and 4) different perspectives leading to different conclusions. Our findings indicate that requirements prioritization more or less includes each of these challenges and is thus a very hard decision for practitioners to make.

The challenges associated with requirements prioritization in our study were manifold and seem to have different types of root causes. We were able to recognize three high-level categories of challenges in requirements prioritization.

Information-combining-related challenges refers to the challenges that occur because it is difficult to both define the aspects that should be taken into account in the decision-making and actually take them into account. These kinds of challenges also seem to be the best understood and reported group of challenges in the literature (e.g. Bubenko 1995; Carlshamre 2002). For example, Regnell et al. (2001) found that it is not easy for decision-makers to define the aspects on the basis of which prioritization decisions should be made.

People- or resource-related challenges refers to those challenges in the prioritization that occur because of the nature of product development work in an industrial setting. We found that involving people in the prioritization process may be difficult for many reasons, for example if the incentives do not support working towards common goals. In addition, there is usually very little time to analyze a single requirement in practice since requirements management systems are overloaded with requirements. Furthermore, requirements prioritization seems to be an ambiguous concept that people in organizations understand differently. These kinds of challenges have not been properly recognized and faced in the earlier studies. For example, many of the requirements prioritization methods introduced in the literature (e.g. Karlsson and Ryan 1997; Wiegers 1999) seem not to take any of these practical limitations into account effectively.

Communication- or description-related challenges refer to challenges that occur because of the difficulties of communicating and describing priority information in the organization. Regnell et al. (2001) touched on these issues by introducing one possible way to illustrate differences between the importance of different requirements for different stakeholder groups. The focus in the literature, however, seems usually to be on

prioritizing a set of requirements, not in how to communicate the results informatively or how to work on the basis of the results.

6.3 Suitability of requirements prioritization methods

According to our study, one possible benefit of using requirements prioritization methods is that practitioners acquire a more controlled way of taking different viewpoints into account during prioritization. Even rough considerations of, for example, user preferences and implementation costs may provide help in decision-making. It seems that the benefits of using a systematic prioritization method go beyond just getting a priority list as an outcome. Similarly, Karlsson et al. (1998) discuss positive side-effects occurring during their pair-wise comparison sessions, such as identifying ambiguous requirements.

However, the practitioners in our cases faced practical difficulties with the methods when evaluating them. For example, in order to prioritize requirements with the methods, the practitioner has to evaluate factors such as “value” or “cost”. In our cases, it was not clear to practitioners what information they should base their evaluations of these factors on. Our interview results support these findings, since, for example, the value of an individual requirement consists of a combination of a large set of case- and context-specific issues affecting it that might be difficult to take into account. Carlshamre (2002) also reports the limited possibilities that exist of defining in advance the aspects that have an effect on priorities.

One challenge in our cases was that the practitioners seemed to mistrust the results they got by the methods. The experiments conducted by Lena Karlsson et al. (2004) reveal somewhat 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)[3].

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. In the methods, requirements are evaluated alone or compared with each other individually. This leads to lists or diagrams of individual requirements. The number of aspects affecting the requirements priorities seems to expand in market-driven companies, compared to those companies operating in the 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 the project business. In addition, there is usually a fixed number of working hours that can be used for product development in one release. The ability of the evaluated requirements prioritization methods to scale in these situations seems to be quite low.

Furthermore, one interesting finding was that practitioners usually have an experience-based outlook regarding the most important requirements that should be implemented first. However, they seem instead to be longing for help in validating their draft decisions. They want to make sure that they are making their prioritization decisions rationally enough and on the basis of the right aspects affecting the importance of the requirements.

[3] AHP is also the technique that was used in our pair-wise comparison case to calculate the results

In summary, our study indicates that prioritization methods can provide help in putting a set of requirements in order and that the results of the prioritization may work as a basis for discussion. However, there are practical difficulties in the usage of methods and therefore prioritization results should be taken more as being indicative than as an ultimate truth. Our results indicate that an organization may, in some cases, benefit from using a prioritization method through, for example, increased communication. However, the prioritization of requirements seems to be too complex and case-specific to be solved by using some simple and generic algorithm.

6.4 Limitations of the study

Defining the concepts internal validity and external validity unambiguously is difficult, since different authors have provided different definitions and placed emphasis on different aspects of these concepts. In this study, internal validity denotes the property of an empirical study where the result is consistent within its local context (Berander and Wohlin 2003). External validity, on the other hand, denotes the property of an empirical study where the result is generalizable to other contexts (Yin 1994).

In Case Study 1 (Investigation of prioritization practices and challenges), the threats to internal validity included the question of whether we were able to gain a representative and true image from the current practices and practical challenges in the case companies. To increase the internal validity of the study, we used several tactics, especially in the data collection phase. First, we triangulated data sources by selecting more than one interviewee from 5 of 7 case companies. In addition, we selected interviewees so that different stakeholder groups within the companies were represented. Second, we used the triangulation of data collection techniques to improve the internal validity of findings within the organizations. Our data collection techniques included interviewing, document analysis, informal discussions, and observation. In addition, we developed practices co-operatively with the practitioners from the companies in order to get deeper information about their current practices. Thus we were able to relate the information gathered from the interviews to these observations. Third, the study was 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.

However, it might be that we were not able to recognize all the important stakeholders affecting the prioritization and selection of requirements in the case companies. 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 it is their needs that the company tries to satisfy 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.

In addition, it might be that interviews, as a primary data collection technique, had a limited ability to show every important aspect of the current industrial practice to the researchers. 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 the challenges experienced might be quite limited.

The threats to external validity include whether, on the basis of the situation in the case companies, it is possible to draw conclusions that concern software companies operating

in the product business in general. 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 in order to avoid a sampling bias.

However, all the case companies 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 companies creates the potential for a sampling bias. As our case companies are willing to work on process improvement and pay for it, it could be argued that they experience more challenges than those companies which do not perceive the need to improve their practices in the areas in which we do research and consult them.

In Case Study 2 (Evaluation of the requirements prioritization methods), the threats to internal validity include whether we were able to gain a representative image of the pros and cons of using prioritization methods within the case company. To increase the internal validity of the study, we triangulated data sources and data collection techniques. In this case too, our data collection techniques included interviewing, document analysis, informal discussions, and observation. Our research focus was more on the subjective opinions of the practitioners than measuring the objective quality of the outcome that different methods provide. This means that the results concerning the quality of the methods can only be seen as indicative.

However, the main limitations of the study are mostly due to the really small sample of case organizations, case projects, and prioritized requirements. We had only one case company, only two case projects, and only 20 requirements to prioritize in both case projects. All the above-mentioned issues are serious threats to external validity and generalization from such a small sample is difficult as such.

Some other authors have evaluated methods in the laboratory or by having university students as subjects (Berander 2004). There is value in conducting such studies and these studies may lead to more exact results concerning, for example, the scalability of the methods. However, the focus in our research was on understanding the phenomena to be found in using prioritization methods in industrial projects. 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. In addition, we think that in order to understand the suitability of the methods for real industrial projects, the experience-based opinions of practitioners are essential.

7. Conclusion

In this section we present a summary of the research and draw conclusions on the basis of the results. We also suggest directions for future work.

7.1 Summary and contribution

In this study, we investigated the existing mechanisms that software companies use for making decisions concerning the features to be implemented in their products. In addition, we investigated how the requirements prioritization methods from the literature suit market-driven software development and discuss the possibilities of solving product planning challenges by employing a requirements prioritization method.

We interviewed 25 practitioners from 7 software companies in order to understand the existing practices used for implementation decisions in the software companies. In addition, the challenges both in this decision-making and in the practices aimed at tackling these challenges were investigated. Furthermore, we evaluated two requirements prioritization methods, the pair-wise comparison technique (Karlsson 1996) and Wiegers' method (Wiegers 1999), in two industrial case projects. As a longitudinal study, we worked for three years in close co-operation with 2 companies in order to gain a deep understanding of their existing prioritization practices and the challenges involved.

The state-of-practice study shed light on existing practices and challenges in the area of requirements prioritization in the case companies. According to our study, the prioritization practices are mostly informal and prioritization methods are not widely used in the industry. However, there appear to be similar mental models that practitioners use as a basis for their prioritization decisions. The most common hidden idea in the practices is to evaluate implementation costs versus the business value gained by implementing the requirement. In addition, the companies aim to utilize their product development resources evenly as a function of time.

Furthermore, our study indicates that the evaluated requirements prioritization methods may provide help for practitioners. However, our study also reveals that the challenges in requirements prioritization are mainly other than putting a clearly defined set of requirements in order. The practitioners usually have an experience-based outlook about the most important requirements that should be implemented first. Instead of a method for prioritizing their requirements from scratch, they seem to be longing for help in validating their draft decisions. The practitioners seem to want to make sure that they are making their prioritization decisions rationally enough and on the basis of appropriate aspects affecting the importance or urgency of the requirements. In addition, they seem to be searching for practices for communicating priority information through the organization.

However, our findings indicate that requirements prioritization in market-driven software development seems to be such a complex decision-making problem that the organization cannot solve it comprehensively by just employing a method. The amount of aspects affecting requirements priorities seems to expand in market-driven companies, compared to companies operating in the project business. For example, the satisfaction of intermediaries (such as local resellers or consultants) was mentioned as one important aspect affecting priorities, which is not that important in the project business.

Our findings indicate that employing a prioritization method is just part of the solution for the prioritization problem. According to our study, more important issues to be taken into account by an organization include: defining the phases in which prioritization is needed; combining the business and technical viewpoints; involving the right stakeholders in different phases, and communication of the priorities and priority rationales through the units of the organization.

7.2 Future work

Our study provides directions for further investigation. One of the research results of this study is that in order to be able to prioritize requirements, case-specific aspects affecting priorities should be taken into account. Other authors too have produced similar results in their earlier studies. However, it has also been found that these aspects are difficult to define and take into account. Therefore, one of the research challenges in the future is to investigate *how* the companies could use the case-specific aspects affecting their priorities on different levels of product development in a more structured way than today.

On the basis of the results of this study, it can be indicated that requirements prioritization methods solve only a part of the challenge of prioritizing requirements in an industrial setting. However, further investigation concerning which parts of the problem can be solved by using prioritization methods and which parts need some other solutions is needed. In addition, a clearer picture of the different parts of the problem and solution areas would be beneficial.

In addition, it is important to investigate how priority information could be communicated efficiently through the organization during development work. One of our findings was that one of the key challenges in organizations is how to communicate information about priorities to other stakeholders. It is important to support practitioners by defining what information needs to be communicated to which stakeholders and what kinds of practices would be of value.

References

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

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

ARGYSIS, C., PUTMAN, R., and SMITH, D. (1985) Action Science. San Francisco: Jossey-Bass.

AURUM, A. and WOHLIN, C. (2003) The Fundamental Nature of Requirements Engineering Activities as a Decision-Making Process. Information and Software Technology 45(14), 945-952.

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

BECK, K. (1999) Extreme Programming Explained. Upper Saddle River, NJ: Addison-Wesley.

BERANDER, P. (2004) Using Students as Subjects. Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04), IEEE Computer Society, Washington DC, USA, pp. 167 – 176.

BERANDER, P. and WOHLIN, C. (2003) Identification of Key Factors in Software Process Management – A Case Study. Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03), IEEE Computer Society, pp. 316-325.

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

BERANDER, P. and JÖNSSON, P. (2006) Hierarchical Cumulative Voting (HCV) – Prioritization of Requirements in Hierarchies, to be published in International Journal of Software Engineering and Knowledge Engineering (IJSEKE) – Special Issue on Requirements Engineering Decision Support, December 2006.

BOEHM, B. (2003) Value-Based Software Engineering. ACM Software Engineering Notes 28(2), 4.

BOEHM, B., EGYED, A., KWAN, J., PORT, D., SHAH, A., and MADACHY, R. (1998) Using the Win Win Spiral model: A Case Study. IEEE Computer 31(7), 33 – 44.

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

BOSCH, J. (2000) Design and Use of Software Architectures – Adopting and Evolving a Product-Line Approach. Harlow, England: Addison-Wessley.

BUBENKO, J. (1995) Challenges in Requirements Engineering. Proceedings of the 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 Journal 7(3), 139-151.

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

CLEMEN, R. T. (1996) Making Hard Decisions: An Introduction to Decision Analysis, second edition. Belmont, CA: Duxbury Press.

CUSUMANO, M. (2004) The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad. Free Press.

DAVIS, A. M. (1993) Software Requirements: Objects, Functions, and States. Upper Saddle River, NJ:Prentice Hall.

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

DVER, A.S. (2003) Software Product Management Essentials. Tampa: Anclote Press.

FAVARO, J.M. (2002) Managing Requirements for Business Value, IEEE Software, 19(2), 15 – 17.

DE GREGORIO, G. (2000) Technology Management via a Set of Dynamically Linked Roadmaps. Proceedings of the IEEE Engineering Management Society, pp. 184–190.

EBERT, C. Requirements BEFORE the Requirements: Understanding the Upstream Impacts. Proceedings of 13th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, pp. 117- 124.

GILB, K. (2006) Evo - Evolutionary Project Management & Product Development. Book manuscript. http://www.gilb.com. (Accessed 14.8.2006)

GORCHELS, L. (2000). The Product Manager's Handbook – The Complete Product Management Resource, second edition. Illinois, USA: NTC business books.

HARWELL, R., ASLAKSEN, E., HOOKS, I., MENGOT, R., PTACK, K.: What is a requirement? Proceedings of the Third International Symposium of the NCOSE. (1993) 17–24

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. Boston: Harvard Business School Press.

IEEE Std 610.12.-1990 IEEE Standard Glossary of Software Engineering Terminology.

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

JACKSON, M. A. (1995) Requirements and Specifications. Wokingham: Addison Wesley.

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 the Japanese Society for Quality Control, 39 -48.

KAPPEL, T. (2001) Perspectives on Roadmaps: How Organizations Talk about the Future, Journal of Product Innovation Management, 18(1), 39-50.

KARLSSON, J. (1996) Software Requirements Prioritizing. Proceedings of the 2nd IEEE International Conference on Requirements Engineering (ICRE1996), IEEE Computer Society, pp. 110-116.

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

KARLSSON, L. and REGNELL, B. (2005) Comparing Ordinal and Ratio Scale Data in Requirements Prioritisation. Workshop on Comparative Evaluation in Requirements Engineering.

KARLSSON, J., WOHLIN, C., REGNELL, B. (1998) An Evaluation of Methods for Prioritizing Software Requirements. Information and Software Technology 39, pp. 939–947.

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

KARLSSON, L., BERANDER, P., REGNELL, B., and WOHLIN, C.: Requirements Prioritisation: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning. In: Proceedings of Empirical Assessment in Software Engineering (EASE2004), Edinbourgh, Scotland (2004)

KOTONYA, G. and SOMMERVILLE, I. (1998) Requirements Engineering: Processes and Techniques. Chichester, England: John Wiley & Sons Ltd.

KUJALA, S. and MÄNTYLÄ, M. (2000) Studying Users for Developing Usable and Useful Products. Proceedings of 1st Nordic conference on Computer-Human Interaction, pp. 1-11.

LAUESEN, S. (2002) Software Requirements - Styles and Techniques. Harlow, England: Addison-Wesley.

LEFFINGWELL, D. and WIDRIG, D. (2000) Managing Software Requirements - A Unified Approach, Addison-Wesley, Upper Saddle River, NJ.

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

MOISIADIS, F. (2002) The Fundamentals of Prioritizing Requirements. (Web) Proceedings of Test and Automation Conference (SETE2002). http://www.seecforum.unisa.edu.au/Sete2002/ProceedingsDocs/05P-Moisiadis.pdf (Accessed 14.8.2006)

NAMBISAN, S. (2001) Why Service Businesses are Not Product Businesses. MIT Sloan Management Review 43(4), 72-80.

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

NOVORITA, R. and GRUBE, G. (1996) Benefits of Structured Requirements Methods for Market-Based Enterprises. Proceedings of International Council on Systems Engineering (INCOSE) Sixth Annual International Symposium "Systems Engineering Practices and Tools", July 7-11, 1996, Boston, Massachusetts.

PHAAL, R., FARRUK, C., MITCHELL, R., and PROBERT, D. (2002) Starting-up Roadmapping Fast. Research Technology Management 46(2), 52-58.

PENNY, D.A. (2002) An Estimation-Based Management Framework for Enhancive Maintenance in Commercial Software Products. Proceedings of the IEEE International Conference on Software Maintenance (ICSM'02), pp. 122-130.

RAUTIAINEN, K. (2004) Cycles of Control: A Temporal Pacing Framework for Software Product Development Management. Licentiate Thesis, Helsinki University of Technology, 2004.

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

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

ROYCE, W.W. (1970) Managing the Development of Large Software Systems: Concepts and Techniques. Proceedings of Wescon, pp. 1-9.

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

SAATY, T. (1980) The Analytic Hierarchy Process, McGraw-Hill, Inc.

SAWYER, P. (2000) Packaged Software: Challenges for RE. Proceedings of the Fifth International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ 2000), Springer, pp.137-142.

SCHACH, S.R., 2002. Object-Oriented and Classical Software Engineering, fifth edition. Boston: McGraw-Hill.

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

SOMMERVILLE, I. (1996) Software Engineering, fifth edition. Wokingham, England: Addison-Wesley.

STRINGER, E. T. (1999) Action Research, second edition. Thousand Oaks, CA: Sage Publications.

WIEGERS, K. (1999) Software Requirements. Redmond, WA: Microsoft Press.

WIEGERS, K. (2003) Software Requirements, second edition. Redmond, WA: Microsoft Press.

YEH, A. (1992) Requirements Engineering Support Technique (REQUEST): A Market Driven Requirements Management Process. Proceedings of the Second Symposium of Quality Software Development Tools, IEEE Computer Society Press, pp. 211-223.

YIN, R. K. (1994) Case Study Research – Design and Methods, second edition. Thousand Oaks, CA: Sage Publications.

ZAVE, P. (1997) Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys 29(4), 315-321.