

S. Barney, A. Aurum and C. Wohlin, "A Product Management Challenge: Creating Software Product Value through Requirements Selection, accepted for publication in Journal of Systems Architecture, 2008.

A Product Management Challenge: Creating Software Product Value through Requirements Selection

Sebastian Barney^{1,2}, Aybüke Aurum¹, Claes Wohlin²

¹*School of Information System, Technology and Management
University of New South Wales, Sydney NSW 2052, Australia
aybuke@unsw.edu.au*

²*School of Engineering Blekinge Institute of Technology,
PO Box 520, SE-372 25, Ronneby, Sweden
sebastian.barney@bth.se, claes.wohlin@bth.se*

Abstract

It is important for a software company to maximize value creation for a given investment. The purpose of requirements engineering activities is to add business value that is accounted for in terms of return-on-investment of a software product. This paper provides insight into the release planning processes used in the software industry to create software product value, by presenting three case studies. It examines how IT professionals perceive value creation through requirements engineering and how the release planning process is conducted to create software product value. It also presents to what degree the major stakeholders' perspectives are represented in the decision making process. Our findings show that the client and market base of the software product represents the most influential group in the decision to implement specific requirements. This is reflected both in terms of deciding the processes followed and the decision-making criteria applied when selecting requirements for the product. Furthermore, the management of software product value is dependant on the context in which the product exists. Factors, such as the maturity of the product, the marketplace in which it exists, and the development tools and methods available, influence the criteria that decide whether a requirement is included in a specific project or release.

Keywords: Value based requirements engineering, product value, business value, project value, requirements prioritization

1 Introduction

Incremental software development is becoming an increasingly commonplace practice among software companies as they have discovered the potential of this approach to reduce the amount of effort and time that needs to be invested in a product development prior to its release (Karlsson et al., 1998; Carlshamre, 2002; Ruhe and Greer, 2003). Examples of organisations that have adopted time-to-market as their principal product development criterion include Ericsson, Nokia, General Electric and Hewlett Packard. The ability for software developers to develop a product that meets customer requirements while offering high value to both their own business and to their customers provides increased reassurance of market success, provided that the product is released at the appropriate time and offers a superior level of quality relative to competitors.

Global competition forces companies to become more competitive and responsive to consumers and market developments. Moreover, rapidly changing market requirements as well as environmental and governmental regulations have stressed the urgency of dramatic changes in software companies for future economic survival. Thus, creating value for software companies is more important than ever before.

Value is created when a company makes a profit. It is critical for a software company to maximize value creation for a given investment. Hence, it is essential to understand the relationships between the technical decisions and the business strategy that drives the value. Boehm (2005a) stated software engineering (SE) is largely practiced in a value neutral setting with every requirement, use case, test case, defect and object being considered equally important. Traditionally, there is a separation of concerns – software developers are confined to turning requirements into verified code. Yet Bullock (2000) found that eighty percent of business value comes from only twenty percent of software components. Furthermore, there is often a mismatch between the decision criteria used by software developers at the organizational level, and the value creation

criteria used by software development organizations (Boehm and Sullivan, 2000). In other words, the alignment of product, project and business decisions is a major problem in the software industry.

The value-based approach in requirements engineering (RE) promotes the alignment of product, project and business decisions (Aurum et al., 2006; Aurum and Wohlin 2007), and the involvement of multiple stakeholders' perspectives in the creation of product, project and business value, while aiming to maximize the value of a release of software through the selection and prioritization of requirements (Wohlin and Aurum, 2005a,b; Biffel et al., 2005). Despite the fact that most release planning literature covers prioritization and dependencies between requirements (Carlshamre, 2002; Karlsson et al., 1998; Ruhe and Greer, 2003), there has been little research into the criteria used in this decision-making process around requirement selection (Wohlin and Aurum 2005a,b; Hu et al., 2006).

This paper reports a continuation and extended version of the work presented in EuroMicro'06 (Barney et al., 2006). The main objective of this paper is to provide insight into the release planning processes used in industry to create software product value. In Barney et al., (2006) we conducted a case study, by examining two products from an Australian company, and investigated the decision making process when creating product value through requirements selection. In this paper, we extend our research to three case studies from one German and two Australian companies. Data is collected from six different products through interviews and questionnaires. The study addresses both the process followed and the criteria used in the decision-making process. It also highlights different stakeholders' roles and their influences on this process. The contribution of this paper is twofold; a) it studies the values applied in prioritization of requirements for different products in one specific company and investigates the release planning process, and b) it examines to what degree stakeholders' perspectives influence the prioritization of requirements in three different companies and then provides a comparison between the companies.

The remainder of this paper is organized as follows: Section 2 covers background knowledge on the value-based approach in software development. Section 3 presents the methodology used in the studies. Section 4 presents the results of data analysis. Section 5 provides a detailed discussion while Section 6 addresses validity threats. Finally, Section 7 concludes the paper.

2 Background

2.1 The Concept of Value

Economic theory defines value at an abstract level, in terms of *use* and *exchange value*. A *use value* is what the customer is willing to pay for the product, and an *exchange value* is the market value of the product (Mill, 1848). 'Value-adding' became very popular in the early 20th century where the focus on product development became the product itself and customer value was seen as being inherent in the product. By the end of the 1980s the product development was focused on the relationships between the customer service and customer needs. Value was created in cooperation with the customer who was seen as an active participant in value creation activities (Storbacka and Lehtinen, 2001; Heinonen, 2004).

The concept of the value-based approach in SE was introduced in late 1990s in the context of decision-making about product lines (Faulk et al., 2000), managing investments in reusable software (Favaro, 2003) and software economics (Boehm and Sullivan, 2000). Since then, the value-based approach has attracted both software practitioners and academics leading to the integration of value considerations into existing and emerging SE principles and practices (Wohlin and Aurum 2005a).

2.2 Defining Value

Value constructions in economic theory are based on customer satisfaction, loyalty and re-purchasing behaviour (Heinonen, 2004). By borrowing the economic theory, we address three aspects of value, namely *product value*, *a customer's perceived value* and *relationship value*. *Product value* is related to the product price and influenced by the quality attributes of the software product. The value of a product increases in direct proportion to its advantage over competitive products or decreases in proportion to its disadvantage (Alwis et al., 2003). A *customer's perceived value* is the benefit derived from the product and is a measure of how much a customer is willing to pay for it, i.e. $perceived\ value = perceived\ benefits/perceived\ price$, where

perceived benefits and price are both measured relative to competing products (Weinstein and Johnson, 1999). A customer's perceived value is influenced by his/her needs, expectations, past experiences, and culture. *Relationship value* is created through the social relationships between the software company and the customer. It exists through the product and customer's perceived value (Hennerber et al., 2005).

A customer views a purchase as a bargain, if *perceived value* > *price*. If *price* > *cost* then the software company will make profit for their sale. The critical success factor for software companies is their ability to develop a product that meets customer requirements while offering high value that provides a certain guarantee of market success (Boehm and Sullivan, 2000; Messerschmitt and Szyperski, 2004). Since the ultimate aim for a software company is to maximize value creation for a given investment, it is essential to understand the relationships between product, project and business level decisions and the business strategy that drives the value (Aurum et al., 2006; Aurum and Wohlin 2007; CIMA, 2004).

2.3 Value Based Approach in Requirements Process

Today software has a major effect on the cost, value and schedule of projects (Boehm, 2005a). However, an organization's success in terms of profitability or market capitalisation does not necessarily correlate with their level of investment in IT (Thorp, 1998). That is because money spent does not always translate into the realisation of benefits. Most studies into the critical success factors, in successful and failed projects, find that the primary critical success factors lie in the value domain. Most projects fail due to a lack of user input, incomplete requirements, lack of resources, unrealistic expectations, unclear objectives and/or unclear timeframes (Boehm, 2005a). This is because projects are tracked by monitoring project cost and schedule (Boehm, 2005b). This approach, unfortunately, does not consider stakeholder or business value. A project can be successful in terms of cost (i.e. if it is finalized within its budget), but may fail to provide any business value. This can be due to not effectively tracking a project when the project plan changes rapidly, there are flaws in user acceptability, operationally the system is not cost-effective or timely market entry is required (Boehm, 2005a).

Favaro (2003) argues that the purpose of the requirements process is to add business value. As global competition forces companies to become increasingly competitive and responsive to consumers and market developments, the emerging discussions in SE indicate that a value-based approach makes all the difference in creating successful product and value for software business because it puts the requirements engineer in the position of managing requirements to make the most strategic opportunities.

Although companies put a great amount of effort into increasing customers' perceived value in the product development process, determining *how* and *when* value is added is still a challenge even in marketing and management science (Gordijn and Akkermans, 2005). This is because value creation strategies are highly contextual (Kotler, 1993) and must be analysed as part of a multidimensional array of variables (Poladian et al., 2003). Unfortunately, there is no prescribed approach to achieving this perception. There are some companies that adopt a single strategy that best suits their circumstances and are quite successful in value creation. Examples of this approach include Sony-Ericsson and Siemens-Nokia marriages for their mobile phone products.

Kotler (1993) states that strategy and product management should change with market demand. Although this is only one of many factors that could influence the relative importance for criteria in the selection of requirements, we were unable to identify a complete list of factors from the literature.

Techniques to reconcile conflicts include requirement prioritization techniques, business case analysis techniques, and stakeholder identification and requirements and negotiation techniques (Boehm, 2005b). It is also important to consider that the value a market attaches to different requirements changes over time (Maurice et al., 2005), but it is not understood how they will change (Wohlin and Aurum 2005a,b). The critical success factor for software vendors is to respond quickly to changing requirements while maintaining a focus on their value proposition, which may, for example, yield a quicker return on investment.

Shaw (2002) adds that features in software only need to be good enough to meet the needs of the users. Recognising that bug fixes come at a cost. The paper highlights that people are able to handle a certain level of unintended behaviour, and making valuable software includes finding this optimal level of quality.

Value-based requirements engineering (VBRE) aims to maximize the value of a release of software to success-critical stakeholders through the selection of requirements (Aurum and Wohlin 2007; Wohlin and Aurum, 2005a,b). VBRE is a very young area of academic study, however, the problem of creating product value through requirements selection and prioritization is real to the development of software products in today's competitive environment. Companies have been forced to change their practices due to current market forces, but there is little theory providing an approach for development of IT intensive solutions that are valuable to all stakeholders (Gordjin and Akkerman, 2003). In this article we examine this topic in industry through three case studies.

2.4 Incremental Development and Release Planning

Incremental software development is a top-down approach to development in which a minimal software product is developed and released in the first increment, and a function(s) or a requirement(s) is added in each successive increment until the product is complete. Each increment or product release may contain all previously elicited requirements in addition to some new requirements or functions allowing the product to cumulatively grow. This approach to software development requires the analysis of requirements, assigning them to increments (Greer and Ruhe, 2004) and releasing each increment with an aim of meeting the expectations and values of stakeholders who are involved in product development.

Release planning is the process of selecting an optimal subset of requirements for realization in a certain release in incremental software development (Carlshamre, 2002). The aim of release planning is to determine the optimal set of requirements, when they should be released and at what cost this should be achieved (Maurice et al., 2005). Researchers agree that release planning is a crucial determinant of the success of the software product as it determines which requirements will go the next release (Carlshamre et al., 2001). Release planning can only be conducted after a product's requirements have been elicited, analysed and specified (Aurum and Wohlin, 2005a). If release planning is done badly it increases risk (Maurice et al., 2005). For example, leaving critical features or difficult tasks to last or ignoring dependencies and interdependencies can result in time and budget overruns and a loss of market share.

Karlsson et al. (1998) suggested that release planning be approached through the prioritization of requirements. Carlshamre et al. (2001) and Dahlstedt and Persson (2003) furthered prioritization by recognising and accommodating the dependencies that exist between requirements. Wiegers (1999) proposed a method of prioritization that recognised and combined benefits and penalties of proposed functionality between multiple stakeholder perspectives, relative costs and relative risks.

2.5 Aligning Release Planning and Stakeholders' Value Perspectives

Value creation in software development is supported by aligning product, project and business level decisions throughout the development process (Aurum et al., 2006; Aurum and Wohlin 2005a,b). Decisions are traditionally made in a client neutral setting as it is easier to assume that all clients have the same or similar expectations in the solution a product will supply (Berry and Aurum, 2005). This does not result in the best solution for all stakeholders. Additionally software developers are increasingly trying to expand the target market for software products (Wohlin and Aurum, 2005; Aurum and Wohlin 2007). Requirements come from a diverse set of stakeholders are not fed into a specific project, but are managed at a product level. Product managers initiate development projects based on a selected set of requirements. We argue that the following value perspectives are important to software developers when creating value:

- **Business perspective:** Business value stems from product sale
- **Product perspective:** Product value stems from customer and market requirements
- **Project perspective:** Project value stems from project budget/timing/delivery etc.

The above value perspectives need to be aligned with product, project and business level decisions made during the software development process. Each stakeholder group's perception of value is different (Keller, 1995; Boehm, 2005b). For example a project sponsor defines value in terms of the cost of the software and the benefits it provides, while a software company measures value in profit. Stakeholders values are often incompatible, and must be reconciled (Boehm, 2005b). Boehm (2005b) argues that:

- Users of the software want many features, while the project sponsor wants to limit cost by minimising the development effort
- Developers want stable requirements, but users want to be able to change requirements
- The system maintainers want their job to be made easier, but the developers and project sponsors want control over the solution provided and the ability to make changes whenever they want

In summary, while product managers are focussing on business perspective and developers are focussing on product perspectives, sales/project managers will be more concerned with project perspective. Interestingly stakeholders are not always aware of their own value propositions, which can often only be elicited through experience in the problem domain (Anderson and Narus, 1998). Additionally only a limited number of companies in the software business are able to define or measure value from their customers' perspectives despite many describing the achievement of this as having "never been more important" (Anderson and Narus, 1998). Maurice et al. (2005) recognised that while iterative development facilitates early customer feedback, allowing faster delivery and a more interactive process; it also creates difficulties with reconciling conflicting stakeholder perspectives.

Stakeholder management, which involves a focus on the relationship between an organisation and its stakeholders over the transactions between an organisation and its stakeholder has found to create and improve the value of the organisation for shareholders (Hillman and Keim, 2001). While taking a more liberal definition of stakeholder management, Berman et al. (1999) found that two aspects of stakeholder management that are most influential on the financial performance of an organisation – employees, and product safety and quality.

Value-based Requirements Engineering (VBRE) exploits the concept of economic value during the RE process (Gordjin and Akkerman, 2003). Boehm (2005a) states that in order to achieve this, VBRE must include practices and principles for a) identifying a system's success-critical stakeholders; b) Eliciting their value propositions with respect to the system and c) reconciling these value propositions into a set of mutually agreed objectives for the system.

3. Methodology

The main purpose of this study is to understand how software product value is created through the RE process, and identify the relative influence of the decision criteria and major stakeholders' perspectives during requirements selection stage of market driven incremental software development. This paper extends the exploratory research conducted in Barney et al., (2006). The following research questions, presented in three groups, are investigated with respect to the collective case study conducted:

- **Value-based approach in release planning**
 - RQ 1 How is release planning conducted in order to create product value?
- **Values influencing requirement selection**
 - RQ 2.1 To what degree do different value propositions influence the selection of requirement for a release of software?
 - RQ 2.2 Does the manufacturer see potential to improve product value by applying the criteria differently in the selection of requirements for a release?
- **Perspectives influencing requirement selection**
 - RQ 3.1 To what degree do the perspectives of the major stakeholders influence the requirements selection and prioritization process in software industry?
 - RQ 3.2 Does the manufacturer see potential to improve product value by changing the level of influence of the major stakeholders?

3.1 Research Model

The research framework in this study is built on Boehm's Theory W model (Boehm 2005a) that proposes a process for VBRE. According to this model, the process follows identification of success-critical stakeholders, elicitation of their value propositions and requirements, and finally reconciliation of value into a mutually agreed set of requirements.

In this study we used the stakeholder groups identified by Wohlin and Aurum (2005a,b) who participate in requirements selection for the next software release. Note that this model addresses stakeholders' perspectives from software company point of view only. These stakeholders group include business, project and product perspectives as illustrated in Figure 1. Each stakeholder group's perception of value is different. For example a project sponsor defines value in terms of the cost of the software and the benefits it provides, while a software company measures value in profit. Based on premises that requirements selection process will be influenced by stakeholders' perception of value, this study aims to investigate the most prominent decision criterion during requirements selection process.

This study used the criteria identified in Wohlin and Aurum (2005a,b). In these studies the researchers aimed to identify a set of orthogonal criteria that influenced the selection of requirements for a release of software. It was determined this was only possible if the criteria were kept at a high level of abstraction. Criteria were identified and confirmed in two main stages: (i) through an in-depth brainstorming session involving three researchers in requirements engineering with industrial contacts, and (ii) the industry-based research participants were asked to confirm the relevance of the criteria identified by the researchers and identify any missing criteria.

Additionally this paper combined both the criteria identified by the Wohlin and Aurum (2005a,b) and the results provided by the participants. All of the criteria originally identified were adopted, with one additional criterion being selected from those identified by the research participants: function is promised/sold. It was felt there was sufficient overlap between the other criteria identified by the participants and those identified by the researcher.

Further this study asked participants to confirm that each criterion was relevant and identify any criteria missing from list provided. The participants were all IT professionals working in the software development industry.

This research is conducted by using three industry-based case studies since case studies are "especially suitable for learning more about a little known or poorly understood situation" (Leedy and Ormrod, 2005). This study employed a mixed research methodology to conduct each case study which is supported by semi-structured interviews, questionnaires and unstructured interviews. Data is collected from two Australian companies (Company A and Company C), and one German company (Company B). Any cultural issues are outside the scope of this study. The researchers of this study do not perceive cultural issues as a factor in this research as both companies are Western companies, with the German company having a multicultural workforce.

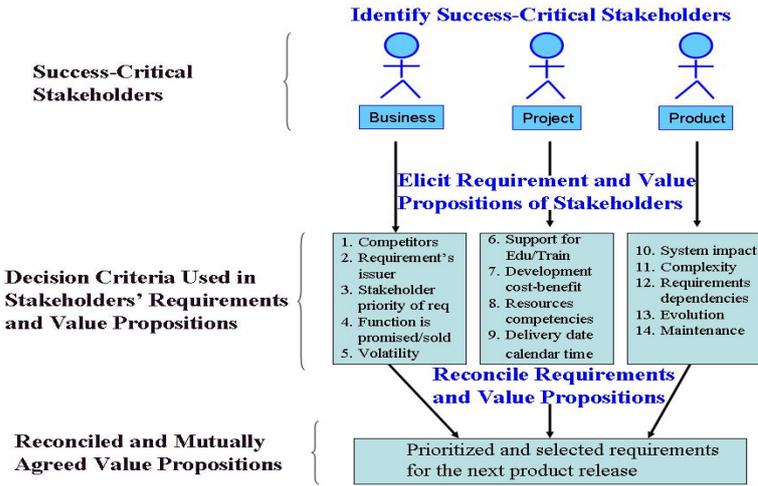


Figure 1 -- Value-based Requirements Engineering Research Model

3.1.1 Semi-Structured Interviews

In phase-1, semi-structured interviews were conducted to understand the processes used in RE and release planning to create product value – research question one. The interview questions (see online supplementary material) covered the interviewee’s background; company’s background; product background; RE process; requirements elicitation process; requirements interpretation, verification and validation process; requirements prioritization and selection; and views on value-based software engineering, and was piloted with academics and professionals within the IT industry.

Each interview went for a period ranging between 25 and 40 minutes. The interviews were held at times suitable for both the researcher and participant. Each participant was given an introduction, providing an overview of the research, and detailing the participant’s rights and responsibilities. Then each participant was asked a series of questions, with the researcher and participant seeking clarification or more information where required. The proceedings of each interview were recorded verbatim electronically. The construction of the interview questions was based on the key activities in the RE, release planning processes and the interviewees’ perception of how to create product value through RE. The interview questions also included company, product and personal (interviewee) background details.

3.1.2 Questionnaire

In phase-2, questionnaires were created to understand the value-system applied to the decision making process in release planning to create product value – research questions two and three. The questionnaire (see online supplementary material) provided the list of criteria presented in this section and asked respondents to identify any criteria missing from the list; mark the criteria relevant to selection of requirements for a release; specify the relative importance of each criteria as they influence the process today; and specify the relative importance of the criteria as the respondent perceived in an optimal situation. The questionnaire was piloted with academics and professionals with in the IT industry.

The questionnaire was targeted at employees who are in a decision-making capacity for the selection of requirements for the product studied. The participants were asked to respond for the product they were identified as being involved with by their company. The company contact at each company selected participants they regarded as appropriate to participate in the questionnaire. The questionnaire focused on the 14 decision criteria which covered three different perspectives, as shown in Figure 1, i.e. business, project and product perspectives.

3.1.3 Unstructured Interviews

In phase-3, the results from the semi-structured interview and questionnaire were then presented to selected participants (including project manager, product manager and development team leader) for their comments in unstructured interviews. The participants were asked to verify and explain the results.

3.2 Case Studies

As illustrated in Table 1, data is collected from three different companies where each company is regarded as one case study. Due to the exploratory nature of this research and limited access to industry, each company was selected using convenience sampling, with a personal with the authors.

Table 1 Descriptions of companies used in the collective case study

Company	Location	Market		Employees	
		Geographic	Industry serviced	R&D	Total
A	Australia	Australia	Telecommunications, finance, government, retail	200	1000
B	Germany	Europe	Telecommunications	Almost	100
C	Australia	Global	Corporate groupware, software development	all	55

Company A

In Company A we studied two products (Product A1 and Product A2) in detail. Product A1 is an enterprise grade document and data repository developed and supported by Company A. The product has been undergoing iterative development for over ten years. It was in its third major release, with the fourth release due months after the study was conducted. Product A2 provided a solution for managing the collection, analysis and processing of paper requests and responses such as election polls, surveys, exams, insurance claims, remittances and general inbound mail. It is developed, hosted and supported by Company A. The product has been undergoing iterative development over two years. At the time of the study the company was preparing for the first major release of this product since it was first released. This product was implemented to replace the existing solutions within the organization that no longer met the requirements of the business.

Company B

Data is collected on only one product from Company B, called Product B, which was a real-time service for mobile telecommunications networks, developed, manufactured and supported. Once implemented this product allows mobile subscribers to decide what each caller hears before the subscriber answers his/her phone call, replacing the traditional ring back tone. The product has been undergoing iterative development for over two years. It has been implemented for over ten mobile telecommunications carriers, and continues to be highly popular with new clients of Company B. Product B needs to be set-up for each mobile telecommunications carrier by Company B. While implementations of this product have a common core, it needs to interface with a number of proprietary systems of the mobile telecommunications carrier. These include the mobile telecommunications network and account management system. Interfaces to these systems, and integration of the service into the existing infrastructure need to be developed and customized. The product has a highly customizable front end for user interaction with the system, the core product software in a middle layer, with a database and specialised telephony switches at the backend. Product B was originally built for a specific client contract, but was designed to allow it to be implemented in other environments.

Company C

We collected data on three products from this company; Products C1, C2 and C3. Product C1 is an enterprise grade issue tracking system developed and maintained by Company C. It is a web-based application with a database backend. The product has been undergoing iterative development for approximately three years. It is used by almost 4,000 organizations in 55 countries. This product can be set-up and integrated by an organization that purchases the software, however, Company C provides support for this product. Product C2 is an enterprise grade wiki developed and maintained by Company C. It is a web-based application with a database backend. This product has been undergoing iterative development for approximately two years. It is used by over 1,800 organizations. This product can be set-up and integrated by an organization that purchases the software, however, Company C provides support for this product. Product C3 is a continuous integration service developed and maintained by Company C. It has just been released for alpha testing. This is used by software developers to build and test code every time it is updated.

4. Data Analysis and Results

Researchers have proposed numerous methods to create software product value through RE and release planning, such as evolve and trade-off analysis techniques. While it is recognized that there is a disparity between formal methods and the processes used in industry, little research has been conducted into the process companies in the software industry use to create software product value.

4.1 Value-Based Approach in Release Planning

This section focuses on decision criteria and release planning applied at the product level based on data from Company A only. The interviews with management level people provided insights to release planning for these two products. We conducted four interviews with people from Product A1 and two interviews from Product A2.

The aim of interview was to understand how the software industry conducted release planning in order to create product value. The interview questions also addressed stakeholder groups, the role of product and requirements engineer, change management issues, and RE activities. The participants from Product A1 were the product manager, R&D managers, IT consultant and System and Program team manager and were the product and R&D managers from Product A2.

Stakeholder groups: The participants identified the clients as the most important stakeholder group in determining what requirements were implemented for a release of software because they represent the revenue stream. However, it was noted that to understand a clients' needs, Company A needed to understand their business and customers. Sales, implementation, operations, development, product management and marketing were also seen as influential groups, but there was disagreement as to the level of influence these stakeholders played.

Product Management: In the past at Company A the sales team dealt directly with the Research and Development (R&D) team to create customer specific versions of Product A1. This created unnecessary rework as similar solutions were redeveloped from scratch, so the role of Product Manager was created to merge these changes into a core product with a strategy to take it into the market place. Additionally both development and sales are biased in their approach to development. The development team is technology focused, the sales team is revenue focused and operations just need to get the job done. One of the participants described how a Product Manager agnostically has a view of all areas and can make a balanced judgment.

The Role of Requirements Engineer: The role of a requirements engineer is not clearly defined in Company A. The Product Manager was responsible for business requirements, while the Development Manager (or research and development team) was responsible for technical requirements.

Change Management: In Company A changes to the product are mainly managed according to their size. One of the participants identified small changes as being rolled out in bug fixes, while bigger changes have a more formal project. The availability of the resources was also identified as the biggest constraint to development.

Requirements Elicitation (for Product Release): Company A held an annual meeting bringing together the sales teams, production team and other vested parties to develop a business strategy for the upcoming year. The objective of this meeting was also to provide *feedback on what are the things that are going to be driving the business that year* to development team. In this meeting, the key products to fulfilling the business strategy would be brought into focus. When a product was brought into focus, it would undergo a requirements elicitation phase involving sales, developers and support teams. The sales team represented general markets requirements and identified product development opportunities. Additionally, industry forms, expos, trade shows and conferences were as ways to keep their product aligned with their competitors, by the marketing and development teams. The requirements elicited from these processes were documented at feature level.

Requirements Interpretation, Verification and Validation: Once business level requirements were defined, a workshop was held with the Product Manager, development team and support team to determine what the requirement elicited means to product in hand. They looked at what must be delivered, what could be delivered and in what timeframe it should be delivered. This workshop was used to develop an understanding of what requirements needed to be implemented. For small discussions phone and email were also used. Since feature descriptions could often be vague business people were invited to meetings. As a result of these meetings the development team determined the system architecture and who would be responsible for what task. Once the functional specification was complete, the client was responsible for verifying and validating that the system specified met their requirements.

Requirements Selection and Prioritization: In Company A, traditionally requirements selection and prioritization has been the role of the R&D Team. Recently, the Product Manager has taken on this role for Product A1. As for Product A2, Product Manager and Development Manager were responsible from this role. Requirements selection and prioritization was a part of the workshop mentioned in the above section. The first stage in requirements selection and prioritization was to ask if the requirements could be implemented. In both products, revenue, product differentiation and sales were the key aims in selecting and prioritizing requirements according to the Product Manager.

The value of requirements from the perspective of Company A and their clients formed part of the requirements selection and prioritization process. One of the participants listed key questions asked as part of this process:

What is the value to (the company)? Is it something that is strategic? Is it eye candy? Is it limited to one particular market place? So we don't only look at the amount of effort required by (Company A), but what it delivers to our customers.

Another participant acknowledged if a client requested a requirement be implemented, but if this requirement was not seen as marketable to other clients, then it would not be rolled into the core product, but a client specific version would be created.

While the input of the workshop played an important role, the final decision as the selection and prioritization of requirements laid with the Product Manager working closely with the Development Manager. Once agreed, the resultant list of requirements was usually made available. As Product A2 was relatively new, a large part of the development effort was rolling customer specific features that have been developed back into the core product as generic components. It was the product manager and development manager's job to start by looking at what should be rolled back into the core product. In order to facilitate an open working environment, the Product Manager wrote a monthly report that detailed all work being done by the R&D team.

Release Planning: Company A has a defined direction for Product A1, but new and existing customers defining new requirements heavily influence this direction. In addition to the formal RE and release-planning project the product evolves with customer specific versions being developed between major releases. If there are features that need to be implemented to win a contract, then they will be made available in either the next release or a customer specific version, in which case they will be rolled back into the core product at a later date. Any requirements that are perceived to have a sufficient market are rolled back into the core product.

The product managers preferred method is to get all of the stakeholders into a meeting to discuss and analyze the business requirements. This process was described by one the product managers:

So we'll sit down and put together a development brief and answer all of the requirements of what we're developing, how long is it going to take, are we going to build for it, how much are we going to sell it for, who else can use it – so all these other questions will form that brief, we'll submit that to R&D and they'll take a look at it and say yep, it'll take two hours, it will cost this much, cost that much, they'll do the building, all the testing that they need to do and sort-of roll it out for that client and then later on roll it out in a major release.

Company A planned a major release of Product A1 every 18 to 24 months. They would try and include the features selected for inclusion. However, if a release needed to be out by certain date, then Company A would make a major release and follow it with a minor release. Although Company A normally worked on a two-year cycle for major releases of its software products, as Product A2 was relatively young, it was currently undergoing six monthly releases. As the time gets closer to a release date, Company A would renegotiate what would be in the release. If it was decided that certain features could not make the desired release date, an additional release with the postponed functionality would be planned to follow.

4.2 Values Influencing Requirement Selection

The objective of this section is to capture and understand any differences on the product level when it comes to which criteria that are important when selecting which requirements to include in a release. The questionnaire allowed us to capture the decision criteria used in requirements selection for Product A1 Product A2, Product B and at Company C. The return rate for the questionnaire s is described in Table 2.

Table 2 Questionnaire return rate

Product/Company	Distributed	Returned	Return rate
Product A1	7	7	100%
Product A2	7	5	71%
Product B	16	9	56%
Company C	40	5	20%

In their study in Sweden, Aurum and Wohlin (2005a,b) identified thirteen criteria, which influenced the selection and/or prioritization of a requirement covering the main stakeholders. While our pilot studies identified several more additional criteria, only one was substantially different to the others. This was the criteria “Function is promised/sold”. Hence, this study used fourteen criteria, based on the response from the participants as illustrated in Figure 1. First the respondents were asked to identify any additional criteria not listed. Then the respondents were asked to mark the criteria they felt were relevant in deciding whether to include a requirement in a release or project. Finally the respondents were asked to provide relative weights regarding the importance of the criteria in two sets, currently (*today*) and if the criteria were applied optimally (*future*). The respondents had 1000 points to spend amongst the criteria. A high number of points meant a criterion was important.

Participants, for Product A1 identified three additional criteria as being important in the decision making process in release planning: (i) creation of competitive advantage, (ii) preferred operating architecture, and (iii) adherence to corporate software design parameters. Participants of Product A2 identified two additional criteria as influencing the release planning process (i) non custom application (Resell a solution, save costs); (ii) future financial worth/new business applications.

The researchers decided these were covered in the existing set of criteria provided in the questionnaire.

The results for all case studies clearly showed that some criteria are more influential than others in the selection and prioritization of requirements. For products A1, A2 and B the value propositions of the business perspective are paramount. However, while the most influential criterion for Product C represents the business perspective the next four all represent the product and project perspective.

Some change in the relative importance of the criteria is also perceived beneficial in all cases. There are no criteria that move consistently across all case studies, but there are some notable similarities between products A1 and A2. The participants for both felt that the influence of the criterion function is promised or sold was overly influential today, while not enough consideration was being given to the impact of delivery date/calendar time.

For each case study the order and relative importance of the different decision criteria for “today” and “future” is presented. The “movement” column in Table 1 shows whether a particular criterion moves up or down when comparing today with future. The “perspective” column shows to which perspective each criteria belongs.

4.2.1 Decision Criteria for Product A1 (Today and Future)

The results for Product A1 are presented in Table 3.

Table 3 Relative importance of different criteria for Product A1 today and in future

	Criteria	Perspective	Today %	Future %	Movement
3.	Stakeholder Priority of Requirement	Business	11.5	10.4	-
2.	Requirement's Issuer	Business	11.5	10.3	-
4.	Function is Promised/Sold	Business	9.0	6.7	-5
11.	Complexity	Product	8.9	6.4	-5
1.	Competitors	Business	8.8	8.9	+1
10.	System Impact	Product	7.8	8.8	+1
12.	Requirements Dependencies	Product	7.6	7.3	-
7.	Development Cost-Benefit	Project	7.3	9.8	+5
9.	Delivery Date/Calendar Time	Project	7.3	7.8	+3
8.	Resources/Competencies	Project	4.9	4.4	-3
14.	Maintenance	Product	4.8	5.1	-
13.	Evolution	Product	4.5	5.8	+2
6.	Support/Education/Training	Project	3.4	3.8	-1
5.	Volatility	Business	2.7	4.4	-3

The business perspective of Product A1 has the most significant influence on the selection and prioritization of requirements for inclusion in the software. When the criteria were ranked in order of influence, all four criteria representing business perspective of Product A1 appear in the first five places. Some project perspective criteria were considered much more important than others. Both the development cost-benefit of the requirement and the impact the requirement has on delivery date had percentage values above 9%, while (5), others were below 3.5% in their importance. Three product perspective criteria were considered of higher importance - the complexity of the requirement, the impact on the system, and any requirement dependencies.

The participants for Product A1 optimally saw a tighter distribution of the criteria. In describing the situation today the criteria were distributed over 8.8 percentage points, compared with an optimal distribution over 6.6 percentage points. The optimal application of the criteria remained business perspective focused; however, this area still reported a significant change. The importance of the criterion if the function has been promised or sold fell five places in the ranking of criteria. The development cost-benefit of the requirement, raised five places when the criteria were ranked how the participants would like to see them applied in future. The most significant change in the criteria representing the product perspective was a decrease in the importance of (11), the complexity of the requirement. This criterion fell five places when ranked against how the participants would like to see them applied.

4.2.2 Decision Criteria for Product A2 (Today and Future)

The results for Product A2 are presented in Table 4.

Table 4 Relative importance of different criteria for Product A2 today and future

	Criteria	Perspective	Today %	Future %	Movement
4.	Function is Promised/Sold	Business	14.5	5.8	-6
8.	Resources/Competencies	Project	11.9	10.7	-1
1.	Competitors	Business	11.3	13.9	+2
7.	Development Cost-Benefit	Project	10.1	7.5	-2
3.	Stakeholder Priority of Requirement	Business	8.5	12.3	+3
2.	Requirement's Issuer	Business	8.3	7.8	+1
11.	Complexity	Product	7.1	5.1	-2
9.	Delivery Date/Calendar Time	Project	6.6	10.3	+4
10.	System Impact	Product	6.4	4.0	-4
5.	Volatility	Business	4.5	5.0	-1
12.	Requirements Dependencies	Product	3.1	4.1	-1
14.	Maintenance	Product	2.9	5.6	+4
6.	Support/Education/Training	Project	2.6	5.1	+3
13.	Evolution	Product	2.2	2.7	-

The business perspective of Product A2 has the most significant influence on the selection and prioritization of requirements for inclusion in the software. When the criteria were ranked in order of influence, all four criteria representing the business perspective of Product A2 appear in the first five places. Unlike Product A1, the most important criteria for Product A2 represent both the business and project perspectives when ranked by importance in release planning. Product A2 also differed from the others in the project perspective. The most important issue from the project perspective was the available resources and their competencies. Two product perspective criterions considered of higher importance: the impact on the system, and the complexity of the requirement.

The optimal (future) application of the criteria remained business focused; however, this area reported a significant change in its makeup. The importance of the criteria if the function has been promised or sold, fell six places in the ranking of criteria going from the most important to the least important business perspective criterion. The order of the business perspective criteria was otherwise unchanged, with the status of competitors with respect to the requirement, and the market's priority of the requirement being the most significant in first and second place of all criteria respectively. The project perspective criterion delivery date/calendar time rose four places when the criteria were ranked against how the participants would like to

see them applied. The participants from Company A optimally saw a tighter distribution of the development criteria. In describing the current situation the criteria were distributed over 4.7 percentage points, compared with an optimal distribution over 1.4 percentage points.

4.2.3 Decision Criteria in Company B (Today versus Future)

The results for Product B clearly indicate that some criteria are more important than others in the selection and prioritization of requirements for a release. It is worth noting that three of the criteria have percentage values above 10% and five have values below 5%. The order and relative importance of the different criteria can be seen in Table 5.

Table 5 Relative importance of different criteria for Product B today and future

	Criteria	Perspective	Today%	Future	Movement
4.	Function is Promised/Sold	Business	14.1	9.6	-1
3.	Stakeholder Priority of Requirement	Business	12.1	12.3	+1
2.	Requirement's Issuer	Business	10.1	10.1	-1
9.	Delivery Date/Calendar Time	Project	9.4	9.4	+1
7.	Development Cost-Benefit	Project	9.1	9.1	0
8.	Resources/Competencies	Project	6.6	6.6	-3
1.	Competitors	Business	6.5	6.5	-3
11.	Complexity	Product	5.9	5.9	-3
10.	System Impact	Product	5.6	5.6	+1
12.	Requirements Dependencies	Product	4.9	4.9	-4
14.	Maintenance	Product	4.9	4.9	+5
13.	Evolution	Product	4.6	4.6	+5
6.	Support/Education/Training	Project	3.4	3.4	+3
5.	Volatility	Business	3.0	3.0	-4

The business perspective of Product B has the most significant influence on the selection and prioritization of requirements for inclusion in the software. The three most important criteria all represent the business perspective of Product B; in order these are (4), if the function has been promised or sold, (3), the market's priority of the requirement, and (2), the stakeholder responsible for issuing the requirement, Requirement's Issuer.

Some project perspective criteria were considered much more important than others. Both (7), the development cost-benefit of the requirement and (9), the impact the requirement has on delivery date had percentage values above 9%, while (5), a requirement's volatility and (6), the ability to provide technical support, education and training for the requirement were below 3.5% in their importance.

Product perspective criteria were considered of fairly equal importance, with all criteria representing these groups clustered together in the results between 5.9% and 4.6%.

The results for how the value criteria should be optimally (future) applied in requirements selection and prioritization for Product B indicate that some change would be perceived as being beneficial. The participants from Company B optimally saw a tighter distribution of the criteria. In describing the current situation the criteria were distributed over 11.1 percentage points, compared with an optimal distribution over 7.9 percentage points. The optimal application of the criteria remained business focused; however, the focus within this area has changed. The first two criteria swapped place, with Company B preferring to see (3), the stakeholder's priority of the requirement as the most important criteria over (4), whether the function has been promised or sold. The optimal application of the criteria also saw the project perspective criteria of (9), the impact the requirement has on delivery date, overtake the importance of (2), the stakeholder responsible for issuing the requirement. It is also of interest to note that (6), the ability to provide technical support, education and training for the requirement rose three places, while (8), the available resources and their competencies fell three places in the ranking.

The criteria representing the product perspective became more evenly distributed when applied optimally. Instead of being distributed over 1.3 percentage points, they were distributed over 3.2 percentage points,

despite the general reduction in range. The effect of a requirement on both (14), maintenance and (13), system evolution, jumped five places in the ranking of the criteria, while considerations of (12), requirement dependencies, fell four places in the ranking.

4.2.4 Decision Criteria in Company C (Today versus Future)

The order and relative importance of the different criteria can be seen in Table 6.

Table 6 Relative importance of different criteria in Company C today and future

	Criteria	Perspective	Today (%)	Future %	Movement
3	Stakeholder priority of requirement	Business	11.2	10.7	-1
9	Delivery date/Calendar time	Project	9.8	7.8	-4
7	Development cost-benefit	Project	9.7	8.8	-1
10	System impact	Product	9.2	12.8	+3
12	Requirements dependencies	Product	8.3	5.3	-6
4	Function is promised/sold	Business	7.2	6.8	-1
8	Resources/competencies	Project	7.0	4.3	-5
11	Complexity	Product	6.7	5.4	-1
14	Maintenance	Product	6.7	8.6	+4
6	Support/Education/Training	Project	5.8	6.2	+2
2	Requirement's issuer	Business	5.3	3.3	-1
13	Evolution	Product	5.2	10.6	+9
5	Volatility	Business	4.1	4.1	0
1	Competitors	Business	3.8	5.3	+4

None of the stakeholder groups is prominent in their influence over the selection and prioritization of requirements at Company C. While the most influential criterion represents the view of the business perspective, the next two criteria represent project perspective criteria, followed by two product perspective criteria. This mixed distribution is continued with the remaining criteria. The results for how the value criteria should be optimally applied in requirements selection and prioritization for Product C1, Product C2 and Product C3 indicate that some change would be perceived as being beneficial. The results can be seen in the table above.

The criteria representing the product perspective have all increased in importance, except for (12), requirements dependencies, showing a general feeling that the issues representing this perspective are currently undervalued. Of most note was (13), evolution, which moved up nine places. It should be noted that Company C only employs people with technical backgrounds. The developers swap between full time development and support, with the time in support used to determine market requirements. The other big movers were the influence of (1), competitors with respect to a requirement, which rose 4 places; and a decrease in importance on (8), resources/competencies, which fell five places.

4.3 Perspectives influencing requirement selection

The relative influence of the different stakeholder groups in each case can be seen in Table 7.

Table 7 Relative influence of stakeholders in requirements selection (percent)

	Product A1		Product A2		Product B		Company C	
	Today	Future	Today	Future	Today	Future	Today	Future
Business Perspective	45	42	46	40	46	40	32	30
Project Perspective	23	27	33	37	28	29	32	27
Product Perspective	32	31	21	22	26	31	36	43

The results for all cases, when grouped by stakeholder group, clearly indicate that some stakeholder groups are more important than others in the selection and prioritization of requirements for a release. The results for

how the value criteria should be optimally applied in requirements selection and prioritization for all products indicate that some change would be perceived beneficial.

The business perspective is the most influential in all cases, both today and in the future, except for Company C. Additionally in all cases the participants felt that the business perspective was overly influential to the detriment of other perspectives.

The product perspective is second most important in selecting and prioritizing requirements for Product A1, with the project perspective is the least important. While the participants in the study felt that optimally, the ranking of these groups would remain the same, it was felt that the opinions of the groups should be valued more equally.

The results for Product A2 indicate that some changes in future would be perceived beneficial when value decision criteria are applied in requirements selection and prioritization. The project perspective is second most important in selecting and prioritizing requirements, while product perspective is least important. The participants in the study saw the release planning being conducted more effectively with a small decrease in the influence of the business perspective and an increase in the influence of the project perspective.

As illustrated in Table 7, project perspective is higher than product perspective for Product A1, while the opposite is true for Product A2. We believe that this difference in requirements selection can be attributed to the respective maturity levels of these two products. This issue is further discussed in Section 5.

Comparison of Perspectives Today

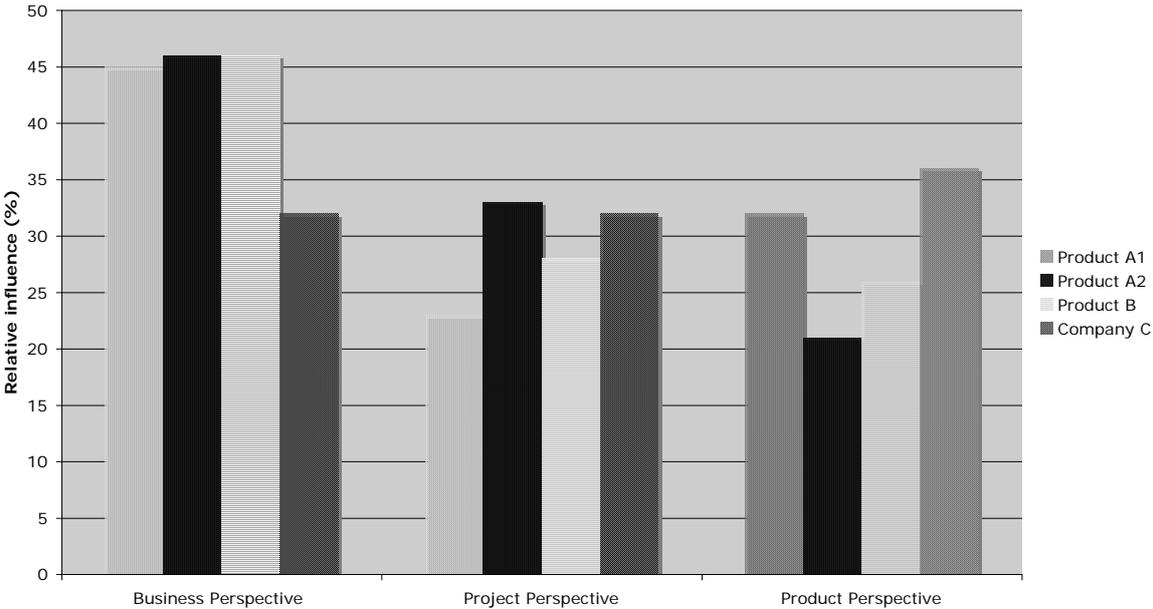


Figure 2 Stakeholders' Perspective at Business, Project and Product Level (Today)

Comparison of Perspective in the Future

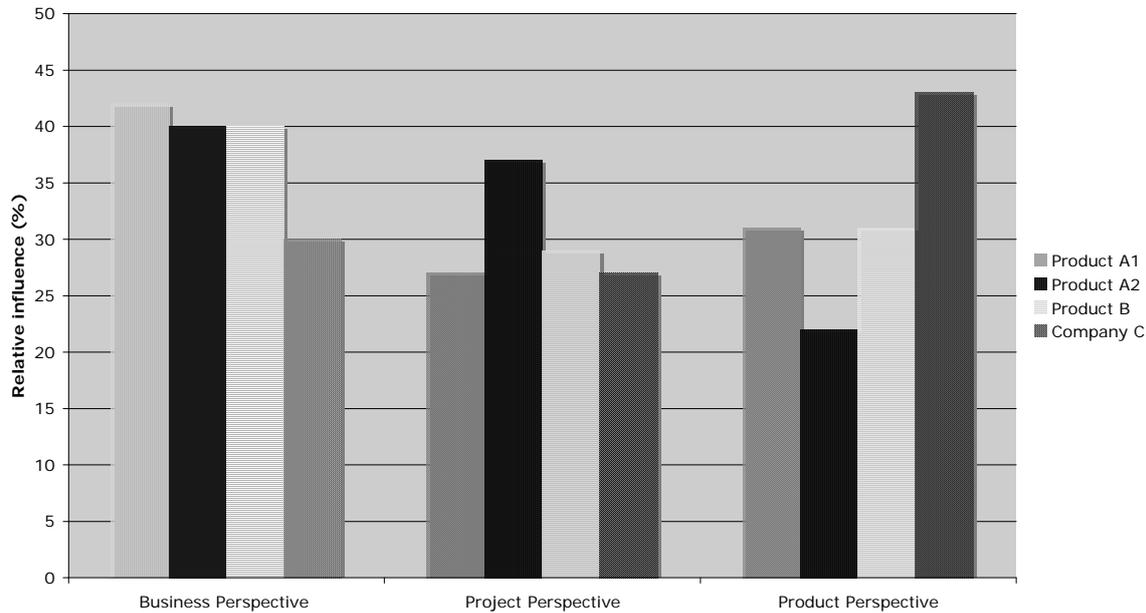


Figure 3 Stakeholders' Perspective at Business, Project and Product Level (Future)

The project perspective is second most important in selecting and prioritizing requirements for Product B, while product perspective is least important. While the participants in the study felt that optimally, the ranking of these groups would remain the same, it was felt that the opinions of the groups should be valued more equally.

In Company C, participants perceived that both business perspective and project perspective are influencing the selection and prioritization of requirements to the detriment of the product perspective. The optimal application of the criteria ultimately saw a decrease in the influence of both the business and project perspective to give greater priority to product related issues. Company C is the only Company in this survey that did not have the business perspective as the most influential group, both today and optimally, in the selection and prioritization of requirements.

5. Discussion of Results

Our results highlighted several interesting aspects of decision making in requirements selection process in incremental development. While between them participants of our study raised the perspectives of each of the key stakeholder groups, it appeared that the value propositions that influenced their decision making process were more intrinsic than part of an explicitly planned process.

5.1 Value-based Approach in Release Planning

Release Planning: Company A in the past developed products for the market without the support of a specific client. This approach was unsuccessful for both clients with the products not meeting the needs of the market and therefore not selling. In order to create software product value, Company A have shifted the focus from fulfilling the perceived value propositions of the market to the much safer known value propositions of clients in a relationship with the company. This "real" client base is a more secure source of revenue for the software development company. There is a relationship between the two organizations and the software development company is creating software to meet the actual needs of the client. Company A moved to a development strategy that requires a client to request a product or feature before it will be developed. This approach created a situation where new features were only made available in customer

specific implementations. This approach created two main problems; new features developed for a customer-specific implementation could not be made available to other clients easily, and when changes were made to the core product many customer-specific implementations need to be released involving a large human effort.

Company A created positions for product managers. The Product Manager for Product A1 and A2 saw his primary role as facilitating the rolling up of customer specific functionality into the core product so that it could be released to the market. An interview participant for Product B expressed a hope that the product manager for Product B would do likewise. A role of the product manager at both companies is to monitor similar products so that the company can react more competitively in bids and marketing material without the need to have functionality implemented. Both companies felt that with a better understanding of the market place sales bids would be more successful.

Company A expressed a need to remain client implementation driven – only implementing requirements when requested by a client. The elicitation of new requirements for a client for all products involves the client and a project manager and/or business analyst from Company A. Where the requirements are technically complex key developers, and in the case of Company A, the Product Manager will also get involved to ensure a common understanding is reached between all of the stakeholders in the requirements. It should be noted that whenever new functionality is to be implemented for Product A1 or Product A2 the client requesting the functionality is used to validate that it is correctly understood and when implemented, that it functions work as required. This ensures that the functionality is meeting the real market need, and not what Company A perceives as the market need.

When a software product has a small client base it is easier to keep the product in line with individual clients' requirements. The client base of Product A2 is both measured in the tens, while the client base of Product A1 is a few hundred. It is much more feasible to create client specific versions of Product A2 with the intention to roll this functionality back into the core product at a later date as the number of customer specific versions of the software can be kept relatively low. This is reflected through the approaches to product development for the three products; RE for Product A1 follows an implementation project lifecycle, whereas Product A2 follow a product release plan except where a client has specific needs that must be met in the short term.

Where development is implementation project driven the requirements selection and release planning are largely defined by the client and will usually be very similar as the client requires the software to meet their needs. However, where software is developed as a product and includes the requirements of many customers the requirements selection and release planning are more easily separated. Dependant on the situation of the Product Manager for Products A1 and A2, issues like time to market are more critical when releasing a product not for a specific customer and it may be desirable to release a version of software that does not fulfil all the requirements with the intention to follow a release with updates. Failing to meet all requirements is unlikely to affect all customers; there are more market opportunities to be considered in this approach.

Developing a software product to individual clients' requirements is beneficial in the early stages as it facilitates the products growth in line with the needs of the client base and the general market. All features that are developed are requirements of the clients, and nothing is developed that is not a requirement of a client. This practice minimises the time to market with a fully featured product, bringing forward revenue generation. The product manager of Product Manager for Product A1 and A2 noted:

It gives us more credibility if we release the product soon; it gives it more functionality, better saleable product.

However, when a company only develops functionality requested, they risk losing competitive advantage where other marketing offerings contain features that have not been requested, but are sufficient to win a contract. In both Products A1 and A2, release planning was very focused towards meeting customer objectives. Customer specific versions of the software acted as a testing ground for requirements and are considered to represent real needs of the market. Selective and controlled rolling of these requirements into the software product core allowed the most suitable requirements to be released to a market that is better understood. The processes followed for both products A1 and A2 involved bringing the success critical stakeholders together so that a common understanding of requirements could be found. The understanding gained then acted as a base for further analysis. The product manager was then responsible for taking all of these perspectives and information on board in making the final decision as to what should be implemented.

We observed the similar approach to product development with Company B in Germany, who also changed their market driven approach to customer specific development. On the other hand, Company C, which was producing to a larger market, (4600 customers across 60 countries) paid more attention to market requirements and their focus on decision criteria in requirements selection was primarily on product perspective, which was different from Companies A and B.

Product Maturity: In two products that we studied at Company A, Product A2 was much younger than Product A1 i.e. Product A1 had been evolving over the past 10 years, whereas Product A2 was much newer, at two years, and it was still trying to gain credibility within the market place. This can be seen through the most important criteria representing this stakeholder group, for Product A1 these were (3) *the stakeholders' prioritization of the requirement* and (2) *the party responsible for issuing the requirement*, while for Product A2 these were (4) *whether the function has been promised or sold* and (1) *the status of competitors with respect to the requirement*. Similarly, (8), *the resources and competencies of development personnel*, were more important for Product A2 than Product A1. As Product A1 was more mature, expertise existed within the company to support this product. However, Product A2 took the company through “unknown waters”. This is inline with the research by Kotler (1993) that says a companies strategy and product management will change with market demands.

Development cost-benefit (7) was also a bigger issue for Product A2 as a lot of money has been spent in its early development and management was more cautious about revenue prospects, whereas Product A1 had become highly profitable for Company A. Looking at the optimal application of criteria, the product manager felt that the *development cost-benefit* (7), for Product A1, was perceived as undervalued due to the time and resources being put into other projects. There was a perception within the organization that the highly profitable Product A1 would lose competitive advantage while resources were focused elsewhere. However, the PM also conceded that there was a greater desire to build functionality with less concern over the revenue it would generate.

The increase in importance of (9), *the impact a criterion has on the delivery date*, for both Product A1 and Product A2 can be explained with reference to how it helps to create a positive market perception when the company is first to bring new functionality to the market place. The massive decrease in the importance of (4) *the functionality being promised or sold* can be explained with reference to the fact that Company A would like to start taking a more structured approach to bringing new functionality into the market place. Rather than developing client specific versions of the software, the company would like to see more controlled release planning, which implements functionality and releases it to the general market.

5.2 Values Influencing Requirement Selection

Decision Criteria in Requirements Selection: The in depth analysis of two products from Company A has provided noteworthy results. Most people were unable to express how value was created through requirements selection and prioritization in their company; however working with customers was seen as the most effective way to make product valuable. The decision criteria and release planning were affected by several factors e.g. the product maturity, customer importance, the size of the customer-base, the services that were provided to customers etc.

5.3 Perspectives Influencing Requirement Selection

Stakeholder Influence in Requirements Selection: The results of the second research question, based on the data collected from three companies, further highlighted the results of the first research question. The business perspective was the most critical group in creating software product value for Company A and B where the product development involved both market-driven and customer specific approaches. As for Company C, the most influential perspective was the product perspective. Although the ranking in decision criteria changed between the products as well as the companies, the overall results clearly pointed out the influence of business perspective in requirements selection, i.e. the first criterion in ranking always related to the business perspective. There was no major disparity between product and project perspectives for Company A and B. The results showed that practitioners were quite aware of the fact that the requirements selection had to be aligned with business strategies which drive the value creation for their companies.

Overall, the results showed that there was no significant differences between what practitioners were applying *today* and how they would like to see it in *future* in relation to requirements selection. In both companies A and B, the business perspective came out as the most influential perspective overall (today and future). However, there was a slight push down to the business perspective and push up in both project and product perspectives in future criteria selection. On the other hand, the product perspective was the most influential perspective in requirements selection for Company C (today and future), however, there was a slight push up for project perspective in future criteria selection. Furthermore, in future criteria selection, the product and project perspectives were more evenly distributed. This indicates that software practitioners accept the importance of the business perspective in requirement selection; however, in future, they would like to see the product perspective receiving more attention during requirements selection.

Customer importance: Just as different stakeholder groups exert different amounts of pressure on the selection and prioritization of requirements, not all customers are equal in the level of influence. Company B identified that some customers are strategically more important than others to the software development company. The factors identified that determine whether a customer is strategically important include the prestige, credibility and marketability of having a particular high profile customer; the ability for larger clients to pay higher premiums, and in the case of multinationals, there is a potential for repeat work in many countries with similar requirements. For these reasons, the software development company is more likely to accommodate changes to the base system as part of the contract for strategically important customers.

For small customers [Company B] won't make a lot of customizations

... we will ... force them down a track that is easier for our developers to do. But if they are a really big customer, either from the respect of paying a lot of money or basically they are important to the company, we will go in there and gather all the requirements come back and work out what it is we can and cannot do and then make a decision based on man-hours and the improvement to the product if it is worth development for that customers.

6. Validity Threats

The following presents the validity threats to the findings, considering four kinds of validity, including conclusion validity, internal validity, construct validity, and external validity.

Conclusion validity: Threats to conclusion validity, are lack of statistical calculations or misuse of statistical assumptions that leads to incorrect conclusions made by the researcher. There may be a risk that conclusions from this study are inaccurate due to low statistical power. We did not use statistical calculations to find patterns in the result. Instead, deductive logic was used because of limited data points. To receive high reliable measures and to avoid poor question and poor layout several pilot studies were conducted. The results of this study must be interpreted with caution as the three companies that we investigated are not a representative sample of the software development industry. The small number of people involved in the decision making process for the inclusion of a requirement in a software system has limited the number of possible responses for each company.

Internal validity: The research instrument in this study was developed with a close reference to literature. In addition, we conducted pilot studies to ensure that the questions relate to the stated objectives of this study. The participants in this study were experienced software practitioners. Two of the participants were involved in both the interview and questionnaire for both Product A1 and Product A2 affecting the internal validity of the results. Each of the products has the same Product Manager and Development Manager. It is possible that these participants applied the same values to product development in general. However, comparing the the Product Manager's responses for the two products showed that more than half of the points were awarded differently both for the criteria today and the optimal application of the criteria, 54% and 59% respectively. The differences for the Development Manager were not as high, with 27% and 30% of the points being awarded respectively. Another potential threat to internal validity is related to the questionnaire. It is always difficult to know whether the respondents have understood the questions as intended and in a similar fashion to one another. This threat was partially address during the third stage of this study, where the results were presented to the product manager in an unstructured interview with the researcher for confirmation.

Construct validity: We asked participants to evaluate the need for further criteria, however it is easier for

the participants to agree with the set of criteria identified by the researchers than disagree. It was easier for the participants to agree with the set of criteria identified by the researchers in advance than disagree, because they knew that the criteria on the list indicated that the researcher considered them as relevant. In addition, it was easier to stick to the stated criteria than proposing new criteria. This is partially taken care of by allowing the participants to assign a relative importance of zero or propose new criteria where they see fit. However, the effect of examining the criteria at a different level of abstraction has not been considered for this study.

External Validity: This threat can cause incorrect conclusions to be drawn from empirical study. There are two threats to external validity that are relevant for this study: interaction of selection and treatment, and interaction of history and treatment (Wohlin et al., 2000). Participants were selected from different geographic locations in Australia and Germany. Both male and female participants were represented. The small sample size was another threat to the external validity of this study. The sample size may affect the conclusions; hence they may not be generalized for the whole SE industry population.

7. Conclusions and Future Study

This article addresses value based approach in requirements engineering when creating product value through requirements selection for a software release. It particularly addresses the issues related to (a) value-based approach in release planning, (b) values influencing requirements selection and (c) stakeholders' perspectives influencing requirements selection.

Interviews at an Australian software development company showed that there is no silver bullet for a value-based approach to release planning. While the interviewees raised the perspectives of each of the key stakeholder groups involved in the development process, it appeared that the value propositions that influenced the selection and prioritisation of requirements were more intrinsic than part of an explicitly planned process.

With respect to the values that influencing requirements selection, VBRE is a new area of study and is still in the theory building stage. Questionnaires at three software companies operating in either Australia or Germany have shown that some criteria are more important than others in the selection of requirements to include in a specific project or release. Moreover, this study has shown that the business perspectives are more influential than the project and product perspectives in general; however this may change due to product and project requirements. The following aspects influenced the requirements selections.

- Maturity of the product
- Requirements source, customer type (big or small – importance: potential for future business, can change a premium on larger companies), contract type
- Size of the customer base – this aspect influences the process more than the result

Furthermore, our findings showed that requirements selection is tightly linked to the business situation and release planning. The requirements selection process for the product was affected by the release data. In other words, in the event that a release is going to be very late, the requirements selection would be different and client business situation would be used to select what they would prefer to release. These findings were particularly true for Company A and Company B where the requirements were defined by specific customers. On the other hand for Company C, where the product is developed based on a wider set of market requirements with many more customers, it was a different story. The selection and prioritization is made by a committee comprised of employees of the software development team representing the views of customers. As the customer base is so large no individual client has the level of influence or control that can be seen in Company A and Company B. The choice is made to provide “the greatest bang for the buck,” to maximize the value for the customer base at large.

With respect to stakeholders' perspectives influencing requirements selection, the following can be said about alignment of product, project and business perspectives:

- It is interesting to see how these perspectives are represented in the entire process. For example, at Company C the business perspective is represented by the developers through their interaction with users in support forums. All developers rotate between support and coding. At companies A and B the

developers are more distant from the requirements and these are generally expressed by the project manager (albeit through the product manager).

- With Company C's approach they like to think of the customers' problem as the developers' problem – because the developer will have to deal with it – so is more likely to correct it, add the feature, etc ... Whereas you do not get this opportunity with the approach by the other companies.
- Because of the way the customers' requirements are presented, they have always been filtered by at least one other stakeholder group's perspective.

These conclusions have some implications. First, as the creation of software product value through requirements selection is not very well understood, it cannot be managed in the most effective way. Greater insight into how value is created through release planning would allow this process to be more effectively managed. Secondly the management of software product value is dependant on the context in which the product exists. Factors such as: the maturity of the product, the marketplace in which it exists and the development tools and methods available influence the criteria that decide whether a requirement is included in a specific project or release. A young product will have a greater need to recoup initial development costs, a more competitive marketplace requires more adaptive development, and standard software tools and techniques speed up development and reduce development resource issues. However, further research is required to determine what aspects of a software product's context influence the decision-making criteria and how these contextual elements influence the priority given to the different decision-making criteria. These issues pose great challenges when it comes to creating software product value through requirements selection.

References

- [1] D. Alwis, V. Hlupic, R. Fitzgerald, Intellectual Capital Factors that Impact of Value Creation, 25th International Conference on Information Technology Interfaces, Croatia, 2003, pp. 411-416
- [2] J.C. Anderson, J.A. Narus, Business Marketing: Understanding What Customers Value, Harvard Business Review, November-December, 1998
- [3] A. Aurum, C. Wohlin, A. Porter, Aligning Software Engineering Decisions. To be published in International Journal on Software Engineering and Knowledge Engineering (IJSEKE) 16(6) (2006) 1-24
- [4] A. Aurum, C. Wohlin, A Value-Based Approach in Requirements Engineering: Explaining Some of the Fundamental Concepts, International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'07), 11-12 Trondheim Norway. Lecture Notes in Computer Science 4542, 2007, pp109-115
- [5] A. Aurum, C. Wohlin, Aligning Requirements with Business Objectives: A Framework for Requirements Engineering Decisions. Workshop on Requirements Engineering Decision Support, REDECS'05, 29 Aug-Sept 2, 2005, Paris, France
- [6] S. Barney, A. Aurum, C. Wohlin, Quest for a Silver Bullet: Creating Software Product Value through Requirements Selection. Proceedings of 32nd EuroMicro Conference on Software Engineering and Advanced Applications (SEAA), Aug 28-Sep 1, 2006, Cavtat/Dubrovnik
- [7] S.L. Berman, A.C. Wicks, S. Kotha, T.M. Jones, Does Stakeholder Orientation Matter? The Relationship between Stakeholder Management Models and Firm Financial Performance. The Academy of Management Journal, Special Research Forum on Stakeholders, Social Responsibility, and Performance, 42(5) (1999) 488-506
- [8] M. Berry, A. Aurum, Measurement and Decision Making. In Value-based Software Engineering by Biff S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, 2005
- [9] S. Biffi, A. Aurum, B. Boehm, H.Erdogmus, P. Grunbacher (Eds.), Value-Based Software Engineering, Springer, Berlin, Heidelberg, New York, 2005, ISBN 3-540-25993-7
- [10] B. Boehm, Value-based Software Engineering: Overview and Agenda. In Value Based Software Engineering, Biffi S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, 2005a
- [11] B. Boehm, Value-based SE: Seven Key Elements and Ethical Considerations. In Value Based Software Engineering, Biffi S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, 2005b
- [12] B. Boehm, K.J. Sullivan, Software Economics: a Roadmap, Proceedings of the International Conference on Software Engineering, 2000, pp.319-343
- [13] J. Bullock, Calculating the Value of Testing, Software Testing and Quality Engineering, May/June 2000, pp56-62
- [14] P. Carlshamre, Release Planning in Market-driven Software Product Development: Provoking an Understanding, Requirements Engineering, 7 (2002) 139-151
- [15] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, J. Natt och Dag, An Industrial Survey of Requirements Interdependencies in Software Product Release Planning, International Symposium on Empirical Software Engineering, CA, (2001) pp. 84-92

- [16] CIMA, Maximizing Shareholder Value -- Achieving Clarity in Decision Making, Technical Report. The Chartered Institute of Management Accountants, UK, 2004
- [17] Å. Dahlstedt, A. Persson, Requirements Interdependencies – Moulding the State of Research into a Research Agenda, Proceedings Fifth IEEE International Symposium on Requirements Engineering (REFSQ'03), Klagenfurt/Velden, Austria, 2003, pp. 71-80
- [18] S.R. Faulk, R.R. Harmon, D.M. Raffo, Value-Based Software Engineering: A Value-driven Approach to Product-line Engineering. 1st Int. Conf on Software Product-Line Engineering, Colorado, August, 2000
- [19] J. Favaro, Value-Based Management and Agile Methods. Proceedings of 4th International Conference on XP and Agile Methods, 2003
- [20] Greer, G. Ruhe, Software Release Planning: An Evolutionary and Iterative Approach. Information and Software Technology, 46 (2004) 243-253
- [21] J. Gordjin, J.M. Akkerman, Value-Based Requirements Engineering: Exploring Innovative e-commerce Ideas, Requirements Engineering, 8(2) (2003) 114-134
- [22] K. Heinonen, Reconceptualizing Customer Perceived Value: The Value of Time and Place. Managing Service Quality, 14(2/3) (2004) 205-215
- [23] S.C. Henneberg, C. Pardo, S. Mouzas, P. Naude, Value Dimensions and Strategies in Dyadic “Key Relationship Programmes”, Dealing with Dualities. Proceedings on the 21st IMP Conference, Rotterdam, 2005
- [24] A.J. Hillman, G.D. Kein, Shareholder Value, Stakeholder Management and Social Issues: What’s the Bottom Line? Strategic Management Journal 22(2) (2001) 125-139
- [25] G. Hu, A. Aurum, C. Wohlin, Adding Value to Software Requirements: An Empirical Study in Chinese Software Industry. 17th Australian Conference on Information Systems, ACIS'06, 6-8 December, 2006, Adelaide, Australia
- [26] J. Karlsson, C. Wohlin, B. Regnell, An Evaluation of Methods for Prioritizing Software Requirements, Information and Software Technology, 39(14-15) (1998) 939-947
- [27] Keller, The Value of Software. Manufacturing Systems, 13(1) (1995) 16 ABI/INFORM Global
- [28] P. Kotler, The Major Tasks of Marketing Management, Marketing Management, 2 (3) (1993) 52-56
- [29] P.D. Leedy, J.E. Ormrod, Practical Research – Planning and Design, 8th Edition, Pearson Merrill Prentice Hall, New Jersey, 2005
- [30] S. Maurice, G. Ruhe, O.Saliu, A. Ngo-The, R. Brassard, Decision Support for Value-based Software Release Planning. In Value Based Software Engineering, Biffl S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, (2005)
- [31] D.G. Messerschmitt, C. Szyperski, Marketplace Issues in Software Planning and Design, IEEE Software, 2004, 62-70
- [32] J.S. Mill, Principles of Political Economy with Some of Their Applications to Social Philosophy, Ashley WJ (Ed), Longmans, Green and Co., London, 1848
- [33] V. Poladian, S. Butler, M. Shaw, D. Garlan, Time is Not Money: the case for multi-dimensional accounting in value based software engineering', Proceedings of the 5th International Workshop on Economics Driven Software Engineering Research (EDSER-5), affiliated with the International Conference in A Software Engineering, 2003,
- [34] G. Ruhe, D. Greer, Quantitative Studies in Software Release Planning under Risk and Resource Constraints, International Symposium on Empirical Software Engineering, CA, 2003, pp. 262-271
- [35] G. Ruhe, M.O. Saliu, The Art and Science of Software Release Planning. IEEE Software, Nov/Dec 47-54 (2005).
- [36] M. Shaw, Everyday Dependability for Everyday Needs, Supplemental Proceedings of the 13th International Symposium on Software Reliability Engineering, 2002, pp. 7-11
- [37] K. Storbacka, J.R. Lehtinen, Customer Relationship Management: Creating Competitive Advantage through Win-Win Relationship Strategies, McGraw-Hill, 2001
- [38] J. Thorp, The Information Paradox, McGraw Hill, 1998
- [39] A. Weinstein, W.C. Johnson, Designing and Delivering Superior Customer Value: Concepts, Cases, and Applications. St. Lucie Press, Boca Raton, FL, USA, 1999
- [40] K.E. Wiegers, First Things First: Prioritising Requirements, Software Development, 7(9) (1999) 48-53
- [41] Wohlin, A. Aurum, Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study. In Biffl S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, 2005a
- [42] C. Wohlin, A. Aurum, What is Important when Deciding to Include a Software Requirement in a Project or a Release? 4th International Symposium on Empirical Software Engineering, Noosa Heads, Australia, 17-18 Nov, 2005b
- [43] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering – An Introduction, Kluwer Academic Publishers, London, UK, 2000

Supplementary Material

The interview questions and questionnaire are available as supplementary material online at <http://www.bth.se/tek/aps/sba/>