STRATEGIC DECISION SUPPORT FOR SOFTWARE INTENSIVE PRODUCT MANAGEMENT

Mahvish Khurum

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2009:08

School of Computing

Strategic Decision Support for Software Intensive Product Management

Mahvish Khurum

Blekinge Institute of Technology Licentiate Dissertation Series No 2009:08

Strategic Decision Support for Software Intensive Product Management

Mahvish Khurum

School of Computing
Blekinge Institute of Technology
Sweden

School of Computing
Publisher: Blekinge Institute of Technology
Printed by Printfabriken, Karlskrona, Sweden 2009
ISBN 978-91-7295-168-6
Blekinge Institute of Technology Licentiate Dissertation Series
ISSN 1650-2140
urn:nbn:se:bth-00444

To my Allah, for blessing me with the abilities and opportunities;
To my mother, for all her support, love and prayers;
To my father, for being there whenever I needed him;

To my brother, for making me smile always;
To my husband, for standing by my side always;
To my Haayed, for being there to add more challenges in an already challenging life ☺

ABSTRACT

Context: At the core of choosing what features and level of quality to realize, and thus offer a market or customer, rests on the ability to take decisions. Decision-making is complicated by heterogeneous understanding of issues such as priority, implication of realization, and interpretations of strategy as pertaining to the short- and long-term development of software intensive systems. The complexity is further compounded by the amount of decision support material that has to be taken into account and the sheer volume of possible alternatives that have to be triaged and prioritized, thousands, or even tens of thousands of requirements can be the reality facing industry. There is a need to develop the functionality that is strategically most important while satisfying customers and being competitive, cost efficient, and minimizing risk. In order to achieve a balance between these factors, it is important that within an organization, all stakeholders agree on the strategic aspects to be considered and their corresponding relative importance.

Objective: The objective of this thesis is to provide strategic decision-making support for product management of software intensive products.

Method: A number of empirical studies, set in both academia and industry have been performed. The research methods used span from systematic reviews, case studies to experiments, all aligned to identify possibilities for improvement, devise solutions, and validate these solutions in several steps.

Result: The methods presented in this thesis can be used to evaluate strategic alignment and identify possible root causes for misalignments. To strengthen strategic alignment, the methods and results in the thesis can be used by product managers for making effective and efficient strategic decisions regarding portfolios and products, following a systematic and aligned process.

Conclusions: The area of product management in the context of market-driven software intensive product development is a field with unique challenges. The specifics of the solutions are based on industry case studies performed to gauge state of the art as well as map the main challenges. The decision-making support developed takes the form of models and methods that support software product management in strategic alignment, requirements triage and portfolio level decisions.

Acknowledgements

First I would like to sincerely thank my supervisor, Dr. Tony Gorschek, for his invaluable feedback, expertise and advice. I especially thank him for continuously ensuring that I get the support I need.

Recognition must also be given to my colleagues in the SERL group for creating a positive, supportive and enjoyable research environment. I would like to extend special thanks to my collaborators: Nino Dzamashvili Fogelström, Sebastian Barney, Robert Feldt, Lefteris Angelis and Dr. Anders Hederstierna, and my co-supervisor, Prof. Claes Wohlin.

I am grateful to everyone who has participated in this research – filling in questionnaires, participating in interviews, providing feedback and analysis, and putting me in contact with the right people. Special thanks must be extended to Ericsson, and in particular my mentors, for their continued support and direction of my research.

This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the project Blekinge Engineering Software Qualities (BESQ).

Table of Contents

Chapter 1 - Introduction

"Nothing is more difficult, and therefore more precious, than to be able to decide."
Napoleon Bonaparte

Everyone has to take decisions in his/her daily life; decisions involving simple everyday routines such as what clothes to wear given the weather conditions, which television set to buy, what food to eat at lunch as well as complex decisions such as the selection of which stocks to invest in. Decision making means that an individual has to choose an action from a finite set of actions $\{a_1, a_2, \ldots\ldots, a_n\}$ where each action can result is a state out of finite set of mutually exclusive states $\{s_1, s_2, \ldots\ldots, s_m\}$. Usually, there are no more than a couple of choices to consider in every day decision-making, such as the selection of a restaurant to go to for lunch or which dress to wear. However, even with limited choices decision making can become difficult when more than one individual are involved, because individual interests/tastes tend to vary, and may even be conflicting.

Furthermore, in case of a variety of choices particularly with reference to a major purchase involving more than one aspect, it becomes difficult to ascertain with confidence, the reliability of the choice made. For example, when buying a new house, it is relatively easy to make a choice based on price alone (one only needs to evaluate which house is the cheapest). However, besides the price, other important considerations such as comfort, locality, and resale options obviously make the choice much harder. Moreover, when different members of a family do not agree on the most important aspects to be considered while buying a house, it becomes extremely difficult to take a decision that is acceptable to all.

Similar decision-making challenges arise when developing software intensive products for mass market, only the scale and complexity are increased by several orders of magnitude, and the effects of the decisions can influence not only one company, but whole markets. The functionality that is most important for the customers might not be as important when other aspects (e.g. price, competition) are factored in. There is a need to develop the functionality that is strategically most important while satisfying customers and being competitive, cost efficient, and least risky. Moreover, it is important that within an organization, all the stakeholders involved agree on the strategic aspects to be considered and their corresponding relative importance.

Software engineering decision support plays a fundamental role in the value generation processes, as it assists in taking the right strategic decisions and developing the right products. Hence, decision support is a crucial component in achieving the goal of delivering value to the involved stakeholders. The processes of determining goals, functions and constraints of a software system are termed as requirements engineering. In market-driven software intensive product development context, software product management is responsible for taking product management and requirements engineering decisions [1, 2]. One of the key elements in making the right decision is to decide and agree on a strategy and roadmap for the development and evolution of a product. Although literature advocates use of product strategies for deciding as what to include in a product [3-6], there has been little research as how this can be done effectively and efficiently. In this thesis focus is put on understanding the challenges faced by software product managers when deciding what to include in a product/product portfolio and addressing some of the challenges by proposing methods that can facilitate decision making.

In the subsequent chapters of this thesis, different research studies are presented and new techniques are introduced. The common theme is that they provide decision-making support for product management of software intensive products. Together, they aim at answering the general research question of this thesis: How can strategic decision support for product management of software intensive products (called software product management from here onwards) be provided?

The first part of the thesis (i.e. Chapter 1) gives an introduction to the research area and the research focus. Section 1 of Chapter 1 describes the evolution of software engineering over time, and the critical role of software product management, along with the challenges is discussed. Section 2 briefly presents related work (detailed related work is presented in each respective chapter). The contribution and outline of the thesis is given in Section 3. A list of included/excluded papers is also given in Section 3 along with the contribution statement of the author in each of the paper. Section 4 outlines the research approach used as well as the research methods utilized. Section 4 also contains the research questions addressed, some basic theories behind the research methodology adopted, along with a discussion about the different research contexts in which the research was executed. Sections 5 and 6 contain validity evaluation and future work respectively. Section 7 summarizes this chapter. The second part of the thesis (i.e. Chapters 2-6) contains the papers included in this thesis.

1. Concepts and Background

This section introduces the key concepts and related work to set the context of research presented in this thesis. Related work specific to the challenges (faced by software product managers) presented in Section 1.2.1 is detailed in Section 2. Detailed background and work related to the research presented in Chapters 2-6 are given in the subsequent chapters.

1.1 Evolution of Software Engineering

Software engineering originally appeared as a response to the software development crisis of the 1960 [7]. It later evolved as an engineering-like response to project failures, serious economic losses, scheduling delays, competitive markets, and increasingly demanding customers expecting more functionality, higher quality and maximum reliability at minimum possible cost. The current state of software engineering is the result of continued advancement as described in the following subsections.

1.1.1 The Era of Technical Software Development

In this era, all efforts were focused on optimizing the technical side of software systems with relatively little emphasis on the business side of the software as part of the development strategy. This can be attributed to the background and expertise of software engineers of that period, who were primarily computer scientists and mathematicians. The business management had little or no involvement in the software development process and the entire process was directed, designed, implemented and evaluated by technical people who focused on automation.

A typical example in this era was to elicit a business problem from a customer (or organization) through interviews, observations or focus groups by the software development team. The role of business representatives was limited to knowing the estimates of the software development activity. If the project overran an estimated schedule and cost, the business management had no power except for expressing dissatisfaction with the service [7].

It was typical that the implemented software created more problems than it solved [7]. For the technical developers, software was seen as a goal rather than as a means to solve business problems for increasing business value that can be measured [7, 8]. In this era, the focus was on information systems (IS) and tailoring a software system to customer needs (bespoke project developments) which were initiated when an organization (or individual customers) came forward with a need or want. The use of software as embedded in products was very limited and not really noticed in the literature or research.

1.1.2 The Era of Traditional Software Engineering

During this era, the development process was still mostly under the complete control of technical developers, but user and business management participation increased to some extent. In order to measure quality as a distance between the actual and desired results, feedback loops were included.

This era still exhibited limited user inputs at the requirements analysis and testing phases [7, 8]. Team structures sometimes allowed business specialists to represent business requirements, but they did not have clearly designated responsibilities, thus minimizing effectiveness of the business involvement. For the technical developers, the major challenge was to deliver a working system with elicited functionality requirements. Measuring the business performance characteristics of the system was completely ignored and testing and evaluation focused on pure technical perspectives of the system [7, 9]. The focus continued to be on bespoke projects.

1.1.3 The Era of Business Evaluation of Software Engineering

Both technical developers and business management started realizing that computers/software did not solve business problems independently as those problems required more than just technical skills. Thus, business management started taking control of software development by adding business value dimension to it. Although technical teams were still technically oriented, their performance was assessed through return-on-investment, net present value, break-even point, risk minimization, customer satisfaction, and added flexibility in the product for the long run [7, 9].

This era also saw the emergence of software economics [10]. Software economics is the field that seeks to enable significant improvements in software design and engineering through economic reasoning about product, process, program, and portfolio and policy issues [10]. However, a serious challenge with software economics is that most software engineers (aka technical developers) are unlikely to understand enterprise-level value creation objectives, and the business management often ignores the fact that the success of a software development effort at the technical level are connected to value creation. In this era, software grew and became a more dominant part as it went from tailored to one customer project to software products being developed for the mass markets [9]. This shift in focus from traditional bespoke software projects to mass market products also increased the need of considering business value perspective when developing a software or software intensive products [9]. However, still most of the research focused on old bespoke and project centered issues and challenges.

1.1.4 The Beginning of Strategic Software Engineering

This era can be categorized as the one with a high degree of collaboration and partnership between the software and business domains in which software products were analyzed and designed in joint sessions [7]. The rationale behind this approach was to create value from diverse needs, backgrounds, and interests (of all the involved success-critical stakeholders) in a well managed, collaborative environment [7, 11]. Thus, a new discipline termed as strategic software engineering (SSE) was born. SSE borrows concepts from economics; management science, engineering, and human resource management [11, 12]:

- Software product lines; domain analysis, core assets and application development
- Software product management decision-making processes
- COTS assessment, acquisition, and management
- Product/project monitoring and control
- Economic models of software development and return-on-investment decisions; effort estimation; value-based software engineering; portfolio management
- Cost/schedule/benefit/quality estimations and processes
- “How much is enough” decisions e.g. how much inspection, testing, evaluation, etc. is enough

Companies in software intensive product development business risk rework, lost market opportunity and market failure due to shortcomings in integrating a strategic perspective into software product management [3]. A strategic decision (contrary to a tactical one) is one that helps plan for and achieve particular product/product family-wide cost, schedule, and quality goals as they are subject to benefit, value, and risk factors [12]. A strategic decision-making perspective is a fundamental foundation of any established business process (software or otherwise), as it aids in binding a context in which rational tactical decisions can later be taken. Rules of thumb, intuition, tradition/experience or gut feeling are no longer applicable to take important decisions about product line creation, evolution or requirements triage and selection for achieving a balance between short-term and long-term gains.

A brief summary of the different eras based on a number of characteristics is given in Table 1.

Table 1 – Characterization of Different Eras

	Era of Technical Software Development	Era of Traditional Software Engineering	Era of Business Evaluation of Software Engineering	Strategic Software Engineering
Duration	1960s to late 1970s	Mid to late 1970s	During the 1980s	During the 1990s and onwards
Technology	Large computers	Databases, networking and communications technology	PC computing	The Internet
Management/ business involvement	Lack of involvement and dependency on technical developers	Restricted to initial systems analysis, system testing and documentation	Beginning of the strategic alignment between business and technical development	Management partners with the development for business value creation
Software development process	The value of software is in the software	Business metrics were implicitly incorporated with the emergence of more disciplined approaches to software development, project management, and quality assurance however still no incorporation of human-driven factors (social and political issues)	Seen as a critical investment in the organization that should be carefully evaluated from a business value perspective	Value creation-driven software process
Natures of Software systems	Large systems and structured problems (transaction-processing applications were common)	Management information systems (MIS) and decision support systems (DSS)	Desktop applications, packages, generic products or organizations and markets	Web-based, wireless, hardware-software combinations, enterprise-driven integration, real time systems
Development Perspective	Bespoke software system development as a project	Still bespoke software system development as a project	Market-driven software products as a series of multiple projects	Software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and are developed from a common set of core assets in a certain prescribed way [refer product lines definitions]
Team structure	100% technical team	Technical but sometimes allowing business analysts involvement	Still technical but monitored by business management and product management. A new set of decisions related to strategies, marketing visions, roadmapping, customer service and support	Interdisciplinary and tailored to the different needs of the various phases of development process.

1.2 Software Product Management

In companies specializing on the development of software intensive products, over the years the role of product manager has emerged to be of strategic value. However, this role and the tasks associated with it have shown to be very complex to carry out [2]. Product management can be defined as a discipline and as a strategic role, which governs a product (or solution or service) from its inception to the market/customer delivery stage in order to generate largest possible value to the business [1]. The product manager is responsible for product requirements, release definitions, product release lifecycles, creating an effective multifunctional product introduction team and preparing and implementing the business cases [1, 2] in a context where many internal and external stakeholders are involved [1, 13]. A product manager can be termed as a ”mini CEO” representing an enterprise or business unit in strategy formulation and strategy realization through operational execution [1]. A product manager works with top management, marketing, sales, engineering, finance, quality and operations to make his products a business success [1]. The success of a product manager lies in balancing projects, people and politics.

In the context of technical sectors with physical products, the domain of product management is well established since the industrial revolution in the 19th century [14-16]. However, only recently software product management has started receiving attention in industry [2] and to a lesser extent in literature [1, 2, 13]. Although several of the existing product management practices [14, 17] can be applied in software product management, specific challenges faced by software product managers can be identified in the context of market-driven software intensive product development. Software intensive products are different from physical products as the manufacturing and distribution of extra copies of software do not require extra costs for the company [2]. Also, existing software intensive products can be customized easily and sold products can be updated using software patches or release updates. However, the downside of these advantages is that the selection/rejection, organization and traceability of incoming requirements becomes very complex [2, 4]. This is further complicated by the fact that many internal and external stakeholders are involved. Moreover, due to the ease of making changes and competitive markets, the software release frequency is relatively high in comparison to non-software products. Furthermore, due to the intangible nature of software products and overlaps in functionality of software products in a product line, the decisions related to product portfolio management are complex to make compared to non-software product. An additional fundamental difference is the fact that during the development of software, the inception (invention), design, prototyping, development, integration, verification and validation, and delivery actions are all included in the “development project”. In case of hardware systems, for example a car, the conception and design stages are set apart from the manufacturing, and all the trial and error is handled separately. The problem is that software development is generally compared to the mature area of manufacturing.

With respect to engineering, insufficient requirements engineering is typically the first sign of product management failure [1]. In the context of market-driven requirements engineering for software product management, a product manager is faced with many decision challenges which are detailed below. These challenges are based on industry experiences reported in literature [2, 18-29] and the author’s personal industrial experience [30, 31].

1.2.1 Challenges for Software Product Management

Below is a list of some important software product management challenges identified in academia and industry.

1.2.1.1 C1: Portfolio Management

Portfolio management entails the decision making about the set of existing software intensive products, about the product lifecycles, establishing partnerships and contracts, introducing new products by looking at market trends, the product development strategy and introducing and managing software product lines [2].

- C1a: Identification of appropriate domain analysis solutions. In order to introduce/adopt the product line concept, one of the major activities is to do a comprehensive domain analysis. Many domain analysis solutions have been presented over the years [33-36]. However, presently SPMs in industry are faced with the challenge of identifying an appropriate domain analysis method, and this is complicated by the fact that evidence of usability and usefulness is rare [37].
- C1b: Does the effort put in to creating and maintaining a product line pay off? For making rational decisions about product line processes and products, it is important to answer the question "Does it pay?" [32]. In order to make sound software engineering decisions it is required to understand the business impact of those decisions and vice versa [32]. Without economic evaluation of adopting product line processes or products, an organization might end up in spending more but getting few or no benefits. Therefore, evaluating the economic viability of product line adoption is critical.

1.2.1.2 C2: Requirements overload

In contrast to traditional requirements engineering, requirements in market driven requirements engineering (MDRE) come from internal sources like developers, marketing groups, sales teams, support groups, bug reports, as well as from external sources such as different users, customers groups from different and multiple market segments, and competitors (through e.g. surveys, interviews, focus groups, and competitor analysis) [5]. The result is a large amount and continuous flow of requirements that threatens to overload the development organization [3]. For a product manager; it is a challenge to handle large amounts of requirements continuously, performing early selection/rejection of requirements, selecting the ones aligned with a specific product's goals and strategy [4, 30].

1.2.1.3 C3: Abstraction Level and Contents of Requirements

Requirements come in different shapes and forms, at different levels of abstraction and stated on varying levels of details. They can range from one liners and goals like statements from for example marketing and sales channels, to detailed technical suggestions from customers. It is challenging for the product management to select from a large number of requirements ranging from technically detailed to quite abstract. Thus, any requirements engineering process targeted for market-driven environment needs to be flexible enough to handle multiple types of requirements.

1.2.1.4 C4: Requirements Dependencies

Requirements dependencies/interdependencies affect primarily requirements selection and release planning. An SPM needs a process/method/tool that enables her/him to take dependencies into account when handling large number of requirements. In industry product managers are faced with the challenge of identifying and planning according to value based dependencies i.e. directly related to customer value and development cost [2, 24]. The value related dependencies are the most important and hardest to identify and plan [24].

1.2.1.5 C5: Creation of Product Value

The analysis and trade-off between requirements dictates long-term vs. short-term product development, as well as the ability to balance functional requirements with non-functional aspects such as architectural longevity and maintainability. Moreover, once analyzed and weighted, the ultimate selection of what requirements to realize, and which to postpone and dismiss, are central to both short-term and long-term success of a product [2]. In this environment key-customer requirements, securing short-term revenues, are often premiered over long-term requirements which are generally associated with higher risk. The same goes for key-customer requirements in comparison with non-functional aspects such as architectural coherence and maintainability, even though the non-functional aspects in the long run might enable savings equal to or greater to the short-term revenues. The ability to balance short-term and long-term requirement selection is paramount, but time-to-market pressure, dominant in market-driven development and pressure for quarterly revenues, often results in prioritizing key-customer short-term requirements. In a market-driven situation, product strategies are the main tool for planning and realizing the goals and objectives of a [4-6, 30]. Thus, from a value creation perspective it is important for product management to evaluate and select requirements that not only create value for key-customers, but also value for the product and the organization by using product strategies [5, 6, 30, and 38].

1.2.1.6 C6: Gap between Marketing, Product Management and Technical Management

Requirements can be viewed as the least common denominator on which decisions regarding what to include in a product's release/offering is decided. As discussed in the previous challenge (C5) that requirements selected for inclusion in a release/offering should be aligned with a product's strategy. This requires: 1) a combination of marketing, strategic and technical perspectives in a product's strategy and, 2) clear and explicit communication of goals and objectives formulated by the business management to the product management group. All groups need to agree on and share *one* vision. For this reason, it is important to assess if all the involved stakeholders are aligned with respect to the understanding of a product's strategy.

1.2.1.7 C7: Market Pull vs. Technology Push

Requirements can be categorized into two main types based on their origin: requirements based on requests/wishes/needs/opportunities identified in the market (termed as Market Pull), and requirements originating from aspirations of creating technical innovations (termed as Technology Push). A strategy should be a mix of strategic, marketing and technical perspectives to assure customer satisfaction by catering for market pull (short-term revenues). Technology push on the other hand in many cases represents long-term investments that are more of a high risk nature which should also be catered for in a product's strategy to ensure long-term revenues and competitive advantage. It is important to create a balance for example; innovative requirements should not selected discarding market requirements and thus leaving the customer unsatisfied. On the other extreme, market requirements should not be premiered over innovative requirements which can results in losing better value in the future.

1.2.1.8 C8: Requirements Prioritization and Release planning

A product's quality is often determined by the ability to satisfy the needs of customers/end-users [39]. In market-driven software development, there are many external and internal sources of requirements (as stated in Challenge C2) which means that prioritization and release planning is harder as conflicting priorities between stakeholders is bound to exist. Product managers are faced with the challenge of planning for a release by balancing the conflicting prioritizations and requirements dependencies (see Challenge C4).

In order to survive in an extremely competitive market with software products that are on time, on budget and on target, software product management has to deal with the above mentioned challenges by incorporating strategic-decision making perspective.

2. Related Work

From the perspective of the challenges presented in Section 1.2.1, it is central to investigate what related work that has been done in order to elaborate on the challenges or in some way address the challenges. This section covers related work in relation to the challenges as well as in relation to the work presented in this thesis.

A number of studies [40-46] have reviewed the state of empirical research in different areas through systematic reviews e.g. computer science, software engineering, web engineering and so on. However, to the best of the author's knowledge no systematic reviews have been conducted to address challenges *C1a* and *C1b* with the focus on usability and usefulness aspects.

In addition, a number of studies have presented solutions to address challenge C2 (*Requirements overload*). For example, Simmon [14] has suggested to measure different aspects of requirements e.g. requirements implementation cost, volatility, schedule slips, lack of domain experience to evaluate a requirement and decide about its selection or rejection. Davis [10] suggests prioritizing requirements based on their interdependencies and estimated effort required, and then performing triage to select a set of requirements that optimizes the probability of product success in its targeted market.

In order to deal with challenges C2 (*Requirements overload*) and C3 (*Abstraction Level and Contents of Requirements*), *Gorschek and Wohlin* have presented a requirements abstraction model called RAM [4]. RAM details action steps as how to specify, place and abstract requirements on product, feature, function and component levels along with the attributes required at each abstraction level. In order to address challenge C4 some work has been done to understand the different types of dependencies between requirements to facilitate identification and planning according to the dependencies [24].

For creation of a product's value to address C5, a number of studies have suggested methods/tools/techniques for achieving, assessing and maintaining alignment [47-54] but none of the studies has assessed/evaluated alignment among involved stakeholders to aid in decisions related to value creation in the context of market-driven software intensive product development. The focus of the earlier studies was on the alignment between business and information systems within an enterprise. Whereas, a market-driven product development company has to look beyond the perspectives of internal business and information systems and projects, and focus on product and company perspectives [6]. This ensures that there is a common understanding of the company's goals and objectives for a particular product.

Research and industry reports put emphasis on the selection of requirements based on product strategies, business goals and the overall vision of an organization as it enables optimizing both long term and short term perspectives as well as aligning the whole organization in the same direction [4-6]. However, none of the studies has presented a solution to utilize product strategies for early decision about requirements selection or rejection. For example, RAM [29] helps in handling large amounts of requirements by abstracting the requirements and enabling triage on the product level requirements. However, RAM does not handle the actual comparison between requirements abstracted to the product level with the product strategies. Vähäniitty [21] suggested "key decision areas" to be incorporated in a strategy to address challenges C5 (*Creation of Product Value*), C6 (*Gap between Marketing, Product Management and Technical Management*) and C7 (*Market Pull vs. Technology Push*). However, description of what factors to be included for each of the decision areas explicitly is rather abstract and how to weigh their relative importance is missing.

A number of researchers have discussed the challenge C8 *(i.e. prioritization and release planning)* and have presented solutions to deal with [55-60]. The contribution of this thesis does not address Challenges C3, C4 and C8. A description of how the chapters in this thesis contribute to deal with rest of the challenges is described in the next section.

3. Contribution and Outline of the Thesis

This section goes through each chapter of this thesis, shortly explaining the contents and contribution, and explicitly linking it to the challenges and related work presented earlier.

Chapter 2 - A Systematic Review of Domain Analysis Solutions for Product Lines

Chapter 2 contains a systematic review of the modeling and scoping activities involved in domain analysis for software product lines from 1998 to 2007. Domain analysis is crucial and central to software product line engineering (SPLE) as it is one of the main instruments to decide what to include in a product and how it should fit in to the overall software product line. For this reason many domain analysis solutions have been proposed both by researchers and industry practitioners. Domain analysis comprises various modeling and scoping activities. The goal of the review was to analyze the level of industrial application and/or empirical validation of the proposed solutions with the purpose of mapping maturity in terms of industrial application, as well as to what extent proposed solutions might have been evaluated in terms of usability and usefulness. The findings of this review indicated that, although many new domain analysis solutions for software product lines have been proposed over the years, the absence of qualitative and quantitative results from empirical application and/or validation makes it hard to evaluate the potential of proposed solutions with respect to their usability and/or usefulness for industry adoption. The detailed results of the systematic review can be used by individual researchers to see large gaps in research that give opportunities for future work, and from a general research perspective lessons can be learned from the absence of validation as well as from good examples presented. From an industry practitioner view, the results can be used to gauge to what extent solutions have been applied and/or validated and in what manner, both are valuable as input prior to industry adoption of a domain analysis solution.

Chapter 2 – main contribution in comparison to the software product management challenges presented in Section 1.2.1 is:

- Presents a systematic review of the modeling and scoping activities involved in domain analysis for software product lines and the results can be used to measure to what extent solutions have been applied and/or validated and in what manner, both are valuable as input prior to industry adoption of a domain analysis solution. Addressing primarily:
    - *C1a: Identification of appropriate domain analysis solution*

Chapter 3 - Systematic Review of Solutions Proposed for Product Line Economics

This chapter contains a systematic review conducted on the papers, which either proposed or reported on experience with economic/cost/cost-benefit/ROI models/methods (called solutions from now on) for software product lines. Since no domain analysis solution can be introduced/adopted by a company without evaluating the economic aspects, the systematic review in this chapter was performed to complement the systematic review presented in Chapter 2. Thus, the motivation was to gauge the level of actual industry adoption, i.e. to what extent the presented solutions are applied and/or validated in industry. In addition to industry validation all other types of empirical results were collected and taken into consideration to offer a detailed summary of the empirical evidence available. To achieve this, the selected papers were categorized and analyzed from several perspectives, such as research basis, method, level of validation and type of empirical results in relation to usability and usefulness of the proposed solutions. For industry practitioners, looking to perform economic evaluations using a certain solution, the results of the study can be used as an indication of maturity as well as to estimate potential risk of a certain solution. From an academic point of view researchers planning studies and evaluation of a solution can use this study as inspiration for study design as the evaluation criteria of the review presented in this paper could be seen as a checklist to test a solution's usability and usefulness. The design of the systematic review is the same as for the one presented in Chapter 2.

Chapter 3 – main contribution in comparison to the software product management challenges presented in Section 1.2.1 is:

- Presents a systematic review of the papers, which either proposed or reported on experience with solutions for software product lines economics and the results can be used to measure to what extent solutions have been applied and/or validated and in what manner, both valuable as input prior to industry adoption of a certain solution. Addressing primarily:
    - *C1b: Does the effort put in to creating and maintaining a product line pay off?*

Chapter 4 - A Method for Early Requirements Triage and Selection Utilizing Product Strategies

Chapter 4 describes a Method for Early Requirements Triage and Selection (MERTS). The method has two main purposes. First, it acts as a stepwise guide to creating product strategies taking both strategic and technical views into account. Secondly, the strategies resulting from MERTS can be used by managers to perform requirements triage, in essence selecting the "right" requirements for realization. The creation of MERTS was preceded by the study of traditional management literature covering strategy formulation, and complemented with exploratory interviews in industry utilizing two technology driven companies working in a market driven context. In addition a set of interviews were conducted with industry managers and technical experts in two organizations to explore important factors affecting the formulation of strategies from an industrial perspective. The factors affecting requirements triage and selection were also explored. With these factors identified, it was concluded that a gap between strategic managers' and technical experts' perspectives existed, as well as there was lack of a structured approach to agree on goals and the ability to perform requirements triage following these goals.

The goal was to develop a method that was usable and useful in a real industry context taking the companies limited time and resources to perform strategy formulation, triage and selection into account.

Following the creation of MERTS the method was initially evaluated by interviewing industry practitioners to get an initial validation of the method's usability and usefulness in industry. During the creation of MERTS a one-size-fit-all philosophy with regards to strategies was not adopted rather by modifying the factors chosen, a "tailoring" towards a product (or organization) can be achieved prior to use in order to adapt to organizational and product specific goals and objectives. This makes it possible for any organization to use MERTS to the degree needed, making it suitable for larger companies as well as small and medium sized enterprises. The contribution of MERTS is to support explicit discussions, and the formulation and documentation of strategies, enabling requirements triage while balancing market pull and technology push.

Chapter 4 – main contribution in comparison to the software product management challenges presented in Section 1.2.1 is:

- Provides support for explicit discussions about product strategies, and the formulation and documentation of strategies, enabling requirements triage and selection

    - *C2: Requirements overload*
    - *C6: Gap between Marketing, Product Management and Technical Management*
    - *C7: Market Pull vs. Technology Push*

Chapter 5 - A Controlled Experiment of a Method for Early Requirements Triage Utilizing Product Strategies

Chapter 5 describes a controlled evaluation of MERTS in a laboratory setting. The study was conducted in an academic setting, using 50 graduate students as subjects.

The main purpose of the evaluation was to assess the effectiveness and efficiency of MERTS in a controlled environment compared to using natural language (NL) formulated strategies, which is the norm in industry. The main motivation of the experiment was to validate MERTS prior to industry piloting as such real industry tests require valuable and hard to obtain resources.

This was achieved by giving subjects 13 product and 18 feature level requirements and asking them to accomplish a considerable amount of work in a relatively short amount of time. The subjects were expected to form an understanding of the concept of product strategy and requirements triage, understand the domain (read and understand the requirements) and then take decisions whether to include or exclude a specific requirement based on the strategy (MERTS or NL) supplied. *Effectiveness* was measured as the number of correct requirements triage decisions and *efficiency* was measured as time taken (in minutes) to perform triage on all requirements. The subjects were offered very little training, and they also possessed little prior knowledge regarding the domain compared to the level of a product manager in industry. Considering these aspects and the total number of correct decisions that resulted in using MERTS we feel that it is safe to draw the conclusion that MERTS is far superior to NL when it comes to strategy formulation and utilization for the purpose of requirements triage. The only potential drawback is that MERTS seems to be more resource intensive to use, although per correct answer we think that MERTS is at least as efficient as the NL option. Moreover, MERTS is essentially a systematic method for thinking and making decisions and that is why it takes more time but avoids errors. This systematic work is missing when using NL strategies.

Chapter 5 – main contribution in comparison to the software product management challenges presented in Section 1.2.1 is:

- By assessing MERTS and the underlying concepts behind the method the validity of the claims made in relation to the method in Chapter 4 are validated.

Chapter 6 - A Method for Alignment Evaluation of Product Strategies among Stakeholders (MASS) in Software Intensive Product Development

Chapter 6 presents a Method for Alignment Evaluation of Product Strategies among Stakeholders (MASS). In a market-driven context, product strategies are the main tool for planning and realizing the goals and objectives of a product [4-6]. Thus, from a value creation perspective it is important for product management to evaluate and select requirements that not only create value for key-customers, but also value for the product and the organization by using product strategies [4, 5]. This implies that good-enough product strategies balancing marketing, management and technical perspectives need to be formulated to enable product management to perform requirements triage, trade-offs, and ultimately requirements selection [30] (MERTS presented in Chapter 4 is one method for formulating product strategies). However, equally important is the alignment between the organization's upper management, the product management and the project (realization) organization, which implies that the overall strategies need to be understood homogenously, and the same strategies need to be the basis for both the planning and the development of a product [30]. This is especially important in relation to the product management organization, as the professionals working within are, through the selection of one requirement over another, the executive arm of upper management, realizing product strategy during the market-driven requirements engineering activities. MASS has been proposed to assess if product strategies are explicit enough to be understood and used by software product management for creating product's value.

Chapter 6 – main contribution in comparison to the software product management challenges presented in Section 1.2.1 is:

- Presents MASS which enables the evaluation of degree of alignment between upper management and the product management with respect to the understanding and interpretation of a product's strategy. MASS shows misalignment, and enables the identification of leading causes.

    - *C5: Creation of Product Value*
    - *C6: Gap between Marketing, Product Management and Technical Management*

3.1 Included Papers

The following five papers are included in the thesis:

1. **A systematic review of domain analysis solutions for software product lines**
   Mahvish Khurum, Tony Gorschek
   In Press: Journal of Systems and Software, 2009, doi:10.1016/j.jss.2009.06.048

2. **A systematic review of papers about economic solutions for product lines**
   Mahvish Khurum, Tony Gorschek, Kent Pettersson
   Published in the Proceedings of 2nd International Workshop on Management and Economics of Software Product Lines (MESPUL 08) in Software Product Lines Conference, 2008
3. **MERTS – A Method for Early Requirements Triage and Selection Utilizing Product Strategies**
   Mahvish Khurum, Khurum Aslam, Tony Gorschek
   Published in the Proceedings of Asian Pacific Software Engineering Conference (APSEC 07), 2007
4. **A Controlled Experiment of a Method for Early Requirements Triage Utilizing Product Strategies**
   Mahvish Khurum, Tony Gorschek, Lefteris Angelis, Robert Feldt
   Published in the Proceedings of the 15th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ09), 2009
5. **A Method for evaluation of Alignment of Strategies among Stakeholders (MASS)**
   Mahvish Khurum, Tony Gorschek
   Submitted to Software Process Improvement and Practice (SPIP), 2009

3.2 Contribution Statement

Mahvish Khurum is the main author for all five papers. This includes the responsibility for running the research process, discussions with the co-authors, and conducting most of the writing. The research in Papers 1and 2 was designed, conducted and reported mostly by Ms. Mahvish Khurum with the assistance of and under the guidance of Dr. Tony Gorschek. The method proposed in Paper 3 was designed and evaluated by the first two authors with the help of Dr. Tony Gorschek, while the writing of Paper 3 was mostly done by Mahvish Khurum and Dr. Tony Gorschek. The design of experiment presented in Paper 4 was performed by Mahvish Khurum together with the co-authors; most of the analysis was done by Lefteris Angelis while the writing and division of work was performed by Mahvish Khurum. The method presented in Paper 5 was designed by Mahvish Khurum with the help of Dr. Tony Gorschek. The study reported in Paper 5 was conducted mostly by Mahvish Khurum and analysis and writing was done by Mahvish Khurum assisted by Dr. Tony Gorschek.

3.3 Papers not Included

Innovative Features Selection using Real Options Theory
Mahvish Khurum and Sebastian Barney
To be published in the Proceedings of Third International Workshop on Software Product Management (IWSPM), 2009

Requirements Management for Continuous Software Product Development
Mahvish Khurum, Sebastian Barney, Nina Dzamashvili Fogelström, Tony Gorschek
Submitted to the *short papers track* in Empirical Software Engineering and Measurement Conference, 2009

Table 2 – Thesis Outline Overview

Chapter Two	A Systematic Review of Domain Analysis Solutions for Product Lines
Chapter Three	A Systematic Review of Domain Analysis Solutions for Product Lines
Chapter Four	A Method for Early Requirements Triage and Selection Utilizing Product Strategies
Chapter Five	A Controlled Experiment of a Method for Early Requirements Triage Utilizing Product Strategies
Chapter Six	A Method for Alignment Evaluation of Product Strategies among Stakeholders (MASS) in Software Intensive Product Development

4. Research Approach and Environment

This section outlines the research questions, research model used for the research and the utilized research methodologies. Section 5.1 details the research questions; Section 5.2 introduces some of the commonly used research methods; and finally Section 5.3 describes the research methods utilized in the research reported in this thesis.

4.1 Research Questions

The main research question posed for the whole thesis in a way summarizes all others. It can be seen more like an overall mission statement or goal, driving all of the following efforts rather than a traditional research question. It is formulated as:

How can strategic decision support for product management of software intensive products be provided?

This overall question binds the field but opens up for an evolution of follow-up questions that need to be answered one by one in order for progress to be made. Subsequent questions were derived from the challenges identified for software product management (see Section 1.2.1) and confirmed through a series of exploratory case studies in industry.

The product line approach is recognized as a successful approach for reuse in strategic software development [61, 62] with the major benefits of product lines adoption reported as reduced time to market [62], reduced cost [63] and improved quality [63, 64]. For these reasons many companies developing software intensive products have either adopted or are considering the adoption of a software product line approach [62, 65]. On product portfolio level it is important to perform domain analysis to identify, capture, and organize the information used in developing software systems within a domain with the purpose of making it reusable (to create assets) when building new products, keeping in view the potential economic benefits. This can help in making informed strategic decisions about reusability of requirements and implemented assets across products in a product family. However, before using a particular domain analysis solution to be used in industry, it is important for software product managers to analyze the level of industrial application and/or empirical validation of all the proposed solutions. The purpose being to map the maturity, as well as to what extent proposed solutions might be proven in terms of usability and usefulness (Challenge C1a). This gave rise to Research Question 1 (RQ1):

> RQ1: What is the strength of the empirical evidence of proposed domain analysis solutions for software product lines?
>
> This can be broken down into the following sub-research questions:
>
> RQ1.1: Are proposed solutions based on needs identified in industry?
>
> RQ1.2: Are proposed solutions applied and/or validated in a laboratory setting or in industry?
>
> RQ1.3: Are the proposed solutions usable?
>
> RQ1.4: Are the proposed solutions useful?

To answer RQ1, Chapter 2 contains a systematic review of domain analysis solutions, where terms such as usability and usefulness are defined.

Moreover, before introducing/adopting a product line approach economic evaluation is a must. Several economic measurement and evaluation models/methods aimed at software product lines have been presented in academia and as industry experience reports. However, in order to gauge the usability and usefulness of the proposed models/methods, it is important to see the empirical evidence of their application and/or validation (Challenge C1b). This led to the formulation of RQ2 which is elaborated upon in Chapter 3.

> RQ2: What is the strength of the empirical evidence of proposed solutions for evaluating software product lines economics?
>
> This can be broken down into the following sub-research questions:
>
> RQ2.1: Are proposed solutions based on needs identified in industry?
>
> RQ2.2: Are proposed solutions applied and/or validated in a laboratory setting or in industry?
>
> RQ2.3: Are the proposed solutions usable?
>
> RQ2.4: Are the proposed solutions useful?

The absence of an explicit and systematic approach for product strategy formulation was identified through exploratory case studies in industry (Challenges C2, C6 and C7). It has also been highlighted in literature that while managers regard strategy formulation and use as the most important aspect of technology management [17], strategy formulation is usually performed on an adhoc basis. In practice, a systematic approach for formulating strategies is often missing [18]. Even if the formulation of strategies is pursued, the factors affecting strategy formulation differ between different stakeholders. Strategic and middle management and technical experts all need to share one vision. Strategic managers often overlook the technical perspective, and technical experts can be unaware of, or overlook, the marketing and strategic manager’s perspectives. This gave rise to RQ3:

RQ3: How can good-enough software product strategies incorporating the strategic, marketing and technical perspectives be formulated?

Chapter 4 presents A Method for Early Requirements Triage and Selection (MERTS), Part 1 of which details steps required to create product strategies while balancing the strategic, marketing and technical perspectives. MERTS also details steps to ensure that product strategies are explicit and that there is a homogenous understanding of product strategies in the organization.

Research and industry reports put emphasis on the selection of requirements based on product strategies, business goals and the overall vision of an organization as it enables optimizing both long term and short term perspectives as well as aligning the whole organization in the same direction [4, 8, 16]. For software product management, it is important to be able to handle large amounts of requirements continuously, performing early triage of requirements, selecting the ones aligned with a specific product's strategy. Since the literature does not illustrate any method as to how can product strategies be used for early requirements triage and selection, this led to RQ4:

RQ4: How can good-enough software product strategies be formulated, incorporating the strategic, marketing and technical perspectives, which can be used for early requirements triage?

Utilizing product strategies for the task of early dismissal/acceptance (triage) of requirements is one of the major parts of MERTS (Chapter 4). The explicit process and steps of MERTS relieve the dependency of an organization on few experienced personnel taking the correct decisions intuitively.

A central part of the validation of research results is the ability to assure applicability in more than one specialized case (e.g. one development organization). Therefore it was regarded critical to ensure that MERTS was tailor-able, making it usable in any organization faced with the challenge of how to select the "right" requirements from the large amount of continuously flowing requirements. In other words, the ability to test the method developed to answer RQ3 and RQ4 in more than one organization, led to RQ5:

RQ5: How can it be assured that research results are applicable in more organizations than the ones primarily involved in the conception of MERTS?

During the creation of MERTS explicit effort was made to ensure that a one-size-fits-all philosophy, with regards to strategies, was not adopted, rather by modifying the factors chosen, a "tailoring" towards a product (or organization) can be achieved prior to use in order to adapt to organizational and product specific goals and objectives. This makes it possible for any organization to use MERTS to the degree needed, making it suitable for larger companies as well as small and medium sized enterprises.

As a part of validating the research it was important to test the central concepts on a larger scale prior to industry trials. The primary motivation was to reduce the risks for industry by doing pre-pilot tests as such real industry tests require valuable and hard to obtain resources. This gave rise to RQ6:

RQ6: How can research results be evaluated prior to industry piloting?

Evaluations in a controlled environment termed as laboratory setting [66] were seen as an option as the resources required were available to the researcher in the form of the academic environment (university and students). This made it possible to devise and execute a controlled evaluation of MERTS for early requirements triage (see Chapter 5). The validity and ethical issues related to using the academic environment as laboratory and utilizing students as subjects have also been discussed in Chapter 5.

Formulating a product's strategy, incorporating different perspectives, is one thing and the primary focus of MERS, however assuring alignment is another. To ensure that the formulated strategy has been explicitly understood and used homogenously while making decisions related to supplying product value demanded further investigation. The degree of alignment between involved stakeholders with respect to the understanding and interpretation of a product's strategy (Challenge C5) needed to be studied and understood. This led to RQ7:

RQ7: How can the degree of alignment, among and between involved stakeholders, with respect to a product's strategic goals and objectives, be evaluated?

Chapter 6 presents A Method for Alignment Evaluation of Product Strategies among Stakeholders (MASS). This was developed in collaboration with industry to enable the evaluation of degree of alignment between stakeholders with respect to the understanding and interpretation of a product's strategy. Furthermore, it not only enables the evaluation of alignment, but also specifically shows misalignment, and enables the identification of leading causes.

4.2 Research Methodology

Research methodologies provide guidelines for researchers as how to avoid bias and subjectivity in investigations. This section contains a set of major methodologies available for empirical research, followed by an analysis of how they relate and are used in the subsequent chapters of the thesis.

Research designs are different ways to establish that the desired phenomenon is observed and influenced as intended [67, 68]. The motivation is to be able to, with as much certainty as possible, separate the affects of the treatment(s) from other influences. Influences can range from an uneven mix of people, to people not working. Research study designs can be broadly categorized into two types: fixed design and flexible design. A fixed design is decided before the study is conducted, whereas a flexible design study can be changed during data collection [67]. Generally, data collected in studies following fixed design is quantitative, while the data collected in studies based on flexible design is qualitative [67].

4.2.1 Fixed Designs

Fixed designs are theory driven, and require specification of treatment and control variables as well as the exact procedure to be followed in advance. This is possible only when there is a substantial amount of conceptual understanding regarding a phenomenon/concept/theory. According to Robson [67] fixed designs can be divided into: true experiments, quasi experiments, single case experiments and non-experimental fixed designs. However, systematic reviews can also be termed as fixed design research.

Using true experiment design a treatment can be compared to a control group or to another treatment group [69]. An investigation is termed as a pure experiment if the subjects are randomly assigned to the different experiment groups. Through true experiments, the effect of a treatment from some other treatment or from the absence of treatment can be clearly determined. However, true experiment may not always be possible due to process dependencies, ethical considerations and so on [69].

Using quasi-experiment design subjects can be assigned to one or more groups based on some non-random criteria. Single case experiment design employs repeated measures on the same subjects before, during and after the treatment.

Non-experimental fixed designs are essentially like true experiment design, however, a certain aspect is not actively changed through applying a treatment [67]. These experiments can be classified into: relational, comparative and longitudinal designs. While in a relational design the relationship between two or more variables is investigated through statistical tests, in a comparative design the main focus is on analyzing the differences between the two groups. In a longitudinal design, trend over a period of time is observed by repeated measures on one or more variables.

A systematic review can be defined as a means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest. Systematic reviews aim to present a fair evaluation of a research topic by using a reliable, rigorous, and auditable methodology [40].

4.2.2 Flexible Designs

Flexible designs are generally used in situations where the theory is being constructed from the perception of an individual or a groups [68] These types of designs are employed in research which is of an exploratory nature. Flexible designs can be divided into case studies, ethnographic studies, and grounded theory research.

In case studies usually the focus is on single case. The data in a case study can be collected in multiple ways and the data collected is mostly qualitative. The limitation of the case study design is that in most cases it is difficult to separate a case from its context, thus limiting the ability to generalize [68, 70].

In ethnographic studies people in their specific context are studied. This type of study employs observation over a longer period of time. The limitation is that the results of the observation are difficult to present in an objective manner [67].

In grounded theory design the focus of the study is on the evolution of a theory which explains what is being observed. The subjects for the study are selected based on a specific purpose to help formulate a theory, either through their behavior or some aspect of them being studied. Data collection in this type of research is continuous. The limitation of this design is that the choice of subjects cannot be made without the researcher having pre-existing theoretical ideas and assumptions, thus potentially biasing the results of the study.

4.3 Research Approach

Looking at the technology and knowledge transfer model presented by *Gorschek et al*. [66] (see Figure 1), there are a number of steps. Step 1 is to identify the problem/issue in industry which can be done using different methods e.g. exploratory case studies, lightweight process assessment or full-scale process assessment e.g. SCAMPI. A number of problems might be identified and there is a need to prioritize them according to the perceived importance and dependency. After problem/issue identification, Step 2 is to study state of the art to formulate the problem and research agenda in close cooperation with industry. Step 3 is to formulate a candidate solution for the identified problem. An important point to consider in this step is that techniques, processes and tools already developed and validated should be utilized if they address the problem identified. The approach should be to build on and refine the research results obtained by others and adding new technologies relevant to the particular problem and context [66]. After the formulation of a candidate solution, a natural progression towards its application in industry calls for some initial validation in academia (Step 4) without expanding expensive industry resources. An initial practical test of the candidate solution in a lab environment can provide fast, valuable feedback, identifying obvious flaws which can be fixed before the actual industry pilot. Once the solution has been refined/updated based on the results of validation in academia, the next step (Step 5) is to perform static validation of the solution in industry. This involves widespread presentation of the candidate solution with the following purposes: getting feedback and suggestions for improvements, validating understanding and coverage, and giving feedback to the practitioners involved in the assessment phase in Step 1. This would enable further refinement of the candidate solution. Step 6 involves piloting refined solution in industry and after getting feedback from the pilot tests, the candidate solution can be updated/refined and release for full scale implementation (Step 6).

Chapters 1 and 2 are systematic reviews of the current literature on domain analysis solutions and economic valuation solutions for software product lines. The research presented in Chapters 4, 5 and 6 follows the technology transfer model presented by *Gorschek et al*. [66].

As suggested by *Gorschek et al*. research must connect to the needs that practitioners perceive as important, or their commitment could be difficult to obtain [66]. MERTS presented in Chapter 4 was developed after evaluating existing literature and assessing current industrial practices using case study approach in two companies (Step 1 and Step 2, see Figure 1). The initial static validation of the candidate solution (MERTS) was performed at the case companies

(Step 4). Prior to industry piloting, MERTS was validated in academia through a controlled experiment (Step 3) using students as subjects which is reported in Chapter 5.

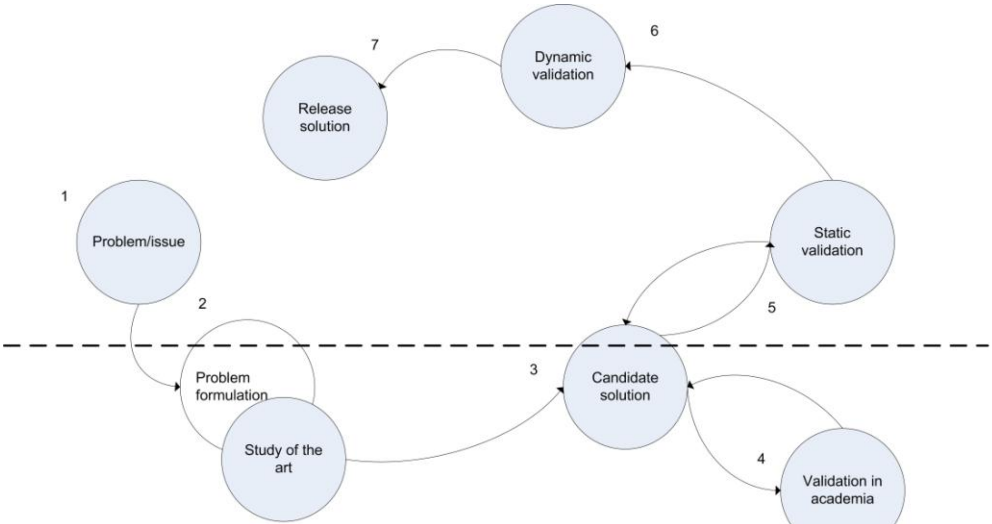

Figure 1 – Overview of research approach and technology transfer model – from [66]

The need for MASS method presented in Chapter 5 was identified through a series of exploratory case studies at the case company (Step 1). MASS was formulated after studying state of the art research literature (Step 2). Already existing established techniques were used as a basis for MASS. Since it is a method to evaluate alignment, it was difficult to test in a laboratory environment with students therefore the method was evaluated using case study research method in the same case company (Step 4).

4.4 Research Methodologies Used

4.4.1 Chapter 2 and Chapter 3: Systematic Review

A literature review is usually the first step in any research and development activity [71]. Through a literature review, a researcher can map existing and previously developed knowledge and initiatives in a particular research field [71]. Due to the growth of scientific production, the role of literature reviews has been proportionally growing larger [71], thus general rules for performing a literature overview have been developed. A systematic review is a specific scientific methodology that goes one step further than the simple literature overviews. It can be defined as a systematic approach to assess and aggregate research outcomes in order to provide a balanced and objective summary of research evidence for a particular topic [40, 72]. The guidelines for performing systematic reviews presented by Kitchenham [40] are: planning the review, conducting the review and reporting the results of the review.

While conducting systematic reviews reported in Chapter 2 and Chapter 3, the step by step guideline provided by Kitchenham [73] was followed. The review design started with the identification of research questions and definitions of relevant terms. The next step was search strategy development which involved identification of search words and search strings along with the research databases to be searched. Once the search strings were identified and search venues were decided, searches were performed and related studies were identified and included/excluded based on a certain criteria. The data extracted from the final set of included studies was classified according to a defined classification scheme. The classified data was then used as an input to answer the research questions

The lessons learned from using systematic reviews are presented below:

- Systematic reviews are a systematic and repeatable approach for assessing and aggregating research outcomes
- It gives an almost complete overview of the studies in a particular research area compared to a simple literature overview which is neither complete nor repeatable
- In order to ensure reliability of a systematic review, more than one researcher should be involved in this process and review protocol should be piloted and updated before starting the actual review
- Unlike the field of medicine, the terminology and quality of context descriptions in software engineering studies varies a lot, and this makes it difficult to use a quality assessment criteria for including studies in the review based on primarily key-words
- It is a lengthy process therefore the scope of the review should be carefully bounded to avoid inclusion of hundreds and thousands of studies for review

4.4.2 Chapter 4: Case Study

Case study can be defined as "*a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon within its real life context using multiple sources of evidence*" [74].

Below the case study ingredients, with respect to the case study reported in Chapter 4, are given:

- *Strategy followed:* qualitative (semi-structured questions) interviews and quantitative questionnaire;
- *Phenomenon studied:* Utilization of product strategy for requirement triage and selection process and initial validation of MERTS
- *Research context:* market-driven software product developing companies
- *Multiple methods:* documentation study and interviews
- *Case companies:* two software product developing companies

The lessons learned from using case study approach for developing and doing initial validation of MERTS are:

- The case study approach helped in identifying the state of practice of product strategies usage and requirements triage practices

- Initial input from industry helped in refining and tailoring MERTS to the industrial context
- Through two cases, it was possible to achieve analytic generalization
- The case study approach was particularly helpful as the phenomenon under study was context specific i.e. utilization of product strategies for requirements triage and selection in market driven software product development

4.4.3 Chapter 5: Quasi Experiment

Experiments are particularly used when there is a need to control a situation and manipulate behavior directly, precisely and systematically [69]. Experiments are particularly appropriate to investigate aspects such as:

- Confirm theories
- Confirm conventional wisdom
- Explore relationship
- Evaluate the accuracy of models
- Validate measures.

The steps of experiment include: definition, planning, operation, analysis and interpretation and presentation and packaging [69]. The experiment reported in Chapter 5 has all the design elements of a true experiment but since the sample used is not truly random, it is termed as quasi. While conducting the quasi experiment reported in Chapter 5, step by step guideline provided by *Wohlin et al.* [69] was followed.

The lessons learned from conducting an experiment, to compare the effectiveness and efficiency of requirements triage using MERTS strategy with natural language (NL) strategies, are:

- Experimentation in a controlled environment enabled the initial evaluation of the effectiveness and efficiency of MERTS strategies for requirements triage
- The claims against a solution can be strengthened using results of experiments which can be used as a proof of the solution's usability and usefulness when discussing piloting of solutions with industry
- Controlled experimentation need to be carefully designed to avoid internal and external validity threats

4.4.4 Chapter 5: Case Study

Below the case study ingredients, with respect to the case study reported in Chapter 5, are given:

- *Strategy followed:* qualitative (semi-structured questions) interviews and quantitative questionnaire;
- *Phenomenon studied:* evaluating the degree of alignment among success-critical stakeholders with respect to the product strategies
- *Research context:* market-driven software intensive product developing companies
- *Multiple methods:* documentation study and interviews
- *Case companies:* one software intensive product development company

The lessons learned from using case study approach for developing and doing initial validation of MERTS are:

- Case study approach helped in identifying the need for evaluating the alignment among stakeholders
- Case study was particularly helpful because the phenomenon under study was context specific that is degree of alignment among success-critical stakeholders with respect to a product's strategy in market-driven software intensive product development.

A summary of the research methodologies, collected data type and context in which research was conducted is presented in Table 3.

Table 3 – Overview of the methodologies used in this thesis

Chapter No.	Methodology	Data Collected	Context
2	Systematic Review	Qualitative	Research literature
3	Systematic Review	Qualitative	Research literature
4	Case studies	Qualitative and quantitative	Software companies
5	Quasi experiment	Quantitative	Academic settings
6	Case study	Qualitative and quantitative	Software company

4.5 Research Setting

This thesis presents a number of empirical studies, set in both academic and industrial settings. The three primary settings used for the collection of data are presented in the following subsections.

4.5.1 Anonymous Companies

MERTS, presented in Chapter 2, has been designed and evaluated with the help of two companies. These companies are described here and in the corresponding chapter, but for reasons of confidentiality remain anonymous.

- The first company, A, is based in UK and operates primarily in the UK market. It is a leading mobile media organization offering a convergence of communications, entertainment and information.
- Company B is also a UK based company and has a long-standing reputation in the area of cellular network planning and optimization. It specializes in mobile network design tools, and has many products, but Plus3G is their matured product with a large customer base.

Collecting data from a number of industrial sources provides greater power in the ability to identify and draw more general trends and conclusions. However, this is not true for this research as two companies used are not representative of the software development industry in general. Still there are a number of advantages from gathering data from a number of different sources - this provides greater confidence for identifying more general trends, allowing future work to focus on areas of key interest without having to find a large number of companies able and willing to participate in a general study.

4.5.2 Blekinge Institute of Technology

As suggested by *Gorschek et al.*, [66] prior to full scale piloting (dynamic validation) of a solution in industry it is beneficial to validate the solution in academia, therefore a controlled experiment was conducted to validate MERTS. Experiments in an academic setting can be seen as a help to bridge the gap between ideas and industry being prepared to get involved. However, this does not mean that these studies do not have academic merit or the potential for industrial application.

Data was collected from students enrolled in a research methodology course offered as part of a master's program in software engineering at Blekinge Institute of Technology for the research presented in Chapter 5. The students involved in the research were considered to be well versed in concepts of market-driven software requirements engineering.

4.5.3 Ericsson AB

Ericsson is a world leader in telecommunications, providing a wide range of products and solutions. Products are developed and sold as generic solutions offered to an open market, although customized versions of the products are also developed. Ericsson is the industrial partner for the research presented in Chapters 5. Ericsson was active in shaping the research, providing access to collect data, analyzing the results, and instigating change based on the results. Of Ericsson's diverse product portfolio, one major software-intensive product has been the subject of the research activities presented in this thesis. It is developed within one part of the organization, so the results are specific to this part of Ericsson and should not be considered representative of Ericsson more generally. The products are not named for reasons of confidentiality. The research partnership with Ericsson provides benefits for both academia and industry. Ericsson is able to get an external perspective that is grounded in research to examine areas perceived beneficial to the company; providing the researcher with industry relevant topics, data and results.

5. Validity Evaluation

All the chapters in the thesis face their own set of validity threats (see each chapter for detailed validity threats). An important validity threat that all the studies in this thesis face is the external validity.

The key idea behind a systematic review is to capture as much available literature as possible to avoid all sorts of bias. Thus, the main challenge with the systematic reviews (see Chapter 2 and Chapter 3) is the reliability. The reliability has been addressed as far as possible by involving two researchers, and by having a protocol which was piloted and hence evaluated. If the study is replicated by another set of researchers, it is possible that some studies that were removed in this review will be included and other studies would be excluded. However, it is highly unlikely that these random differences based on personal judgments would change the general results. It may change the actual numbers somewhat, but it is not likely that it would change the overall results as they are dominantly skewed towards one end of the spectrum (see Section 4). Thus, it is concluded that in general we believe that the external validity of the study is high given the use of a very systematic procedure, consultation with researchers in the field and involvement and discussion between the two researchers.

In case of Chapters 4, 5 and 6, external validity threats arise due to the fact that relatively few cases are studied in industry: in the case of MERTS it is only two organizations (see Chapter 4), in case of MERTS evaluation only 50 subjects (see Chapter 5) and in the case of MASS only three products in one organization (see Chapter 6). In order to deal with this threat, one-size-fit-all philosophy was not followed while designing MERTS and MASS. The strategic factors identified through MERTS steps and the evaluation framework used in MASS can be tailored, based on different product strategies used in different organizations.

Moreover, it is important to stress the fact that working with a research partner in industry generally outweighs the potential drawbacks, as industry practitioners add to the value of solutions in a tangible way through their input and feedback. To support this claim further, since the solution is designed on actual needs identified in industry the chances for application in more than one organization increase.

6. Future Research

There is a real need to actively address the issues of providing support to enable software product management to deal with the strategic decision making challenges effectively.

From the systematic reviews presented in Chapters 2 and 3, there arises a need for future research as how the level of empirical application and industrial validation for the solutions presented for domain analysis and economic evaluation of introducing/adopting software product lines can be improved. The questions and corresponding data extraction categories can be used as a checklist when designing application/validation studies. However, how to do experimentation in software product lines to judge the usability and usefulness of solutions for software product lines analysis and realization still remains a question.

For MERTS (presented in Chapter 4 and evaluated in Chapter 5), an important aspect that needs further research is reduction of subjectivity while strategic factors are identified and given weights. Moreover, currently MERTS do not take in to account requirements dependencies when suggesting steps for triage, this needs further exploration.

If MASS assessment can be replicated in other market-driven software intensive product development organizations, another area with great potential is to formulate a framework with probable solutions against all the identified root causes of misalignments. The framework can look into different characteristics of the root causes and potential solutions and discuss the applicability in different contexts.

7. Summary

This section provides an overview of this chapter. It reiterates the main concepts, relates each of the research contribution with the appropriate chapters and summarizes the findings.

Software Engineering has evolved steadily from its initial days in the 1960s until today. The ongoing goal to improve technologies and practices, seeks to improve the productivity of practitioners and the quality of applications. Previously, software engineering has been presented as a technical process. However, current emerging trends require software engineering in general, and software product management in particular, to be viewed from strategic, business-oriented and interdisciplinary perspectives, of which the technical perspective is but one central view [7].

Companies developing software intensive products risk rework, lost market opportunity and market failure due to shortcomings in integrating a strategic perspective into the management of product development [3]. A strategic decision (in contrary to a tactical one) is one that helps plan for, and achieve, a particular product/product family-wide cost, schedule, and quality goals as they are subject to benefit, value, and risk factors. A strategic decision-making perspective is the fundamental foundation of any established business process (software or otherwise), as it aids in binding a context in which rational tactical decisions can be taken. Rules of thumb, intuition, tradition/experience or “gut feeling” are no longer enough to make important decisions regarding product or product line creation.

The research presented in this thesis stemmed from the challenges faced by software product management while taking different strategic decisions related to software product families and individual products. The individual contribution of each part of the thesis has been explored in previous sections.

An important contribution of this thesis is that all results, whether they are in the form of systematic reviews or methods such as MERTS for requirements triage, or MASS for assessing alignment among stakeholders, can be used by both researchers and practitioners. The detailed results of the systematic reviews (Chapter 2 and Chapter 3) can be used by individual researchers to see gaps in research and opportunities for future work, and from a general research perspective lessons can be learned from the absence of validation as well as from good examples presented. From an industry practitioner view, the results can be used to gauge to what extent solutions have been applied and/or validated and in what manner, both valuable as input prior to industry adoption.

MERTS, presented in Chapter 4, can be viewed as a first method of its kind that presents a way to perform early requirements triage utilizing product strategies. The need for MERTS was identified in literature and confirmed in industry where initial validation was performed. In addition, in order not to expend expensive industry resources, prior to confirming effectiveness and efficiency, a controlled experiment to assess the efficiency and effectiveness was also performed (Chapter 5). The method and its controlled empirical evaluation can be used by researchers for future research. For software product managers in industry it can be used in the context of requirements triage for market-driven software product development. Moreover, the controlled experiment clearly shows the applicability of this method with respect to its effectiveness and efficiency for requirements triage.

Furthermore, to deal with the challenge of creating a product’s value the need for explicit understanding and use of product strategies was identified in industry. MASS presented in Chapter 6 details steps as how to assess the degree of alignment between success-critical stakeholders with respect to the understanding and interpretation of a product’s strategy.

Further, it not only enables the evaluation of alignment, but also specifically shows misalignment, and enables the identification of leading causes.

Another important contribution of this thesis is that the methods presented (MERTS and MASS) were designed keeping tailor-ability and modularity in mind. For example, MERTS can be tailored to suit different organizations depending on their strategies and corresponding factors. Similarly, MASS can be used by large, medium or small organizations to assess the degree of alignment by involving as many success-critical stakeholders as required. Recognition that one-size-does-not-fit-all, and that one solution does not work in isolation but has to be applied in a context, is essential.

The area of software product management in the context of market-driven software intensive product development is a field with unique challenges. The fact that it is possible to validate research results and findings addressing these challenges in industry is an opportunity that should be utilized. This has been recognized by many contributors to the field over the years, resulting in promising and definitely relevant results, many of them used as an inspiration to the research presented in this thesis.

The overall goal of the research presented in this thesis is to provide the software intensive product managers with strategic decision support by presenting solutions for the challenges faced by them in the context of market-driven software intensive product development.

References

[1] C. Ebert, "The impacts of software product management," *Journal of Systems and Software*, vol. 80, pp. 850-861, 2007.

[2] I. v. d. Weerd, S. Brinkkemper, R. Nieuwenhuis, J. Versendaal, and L. Bijlsma, "Towards a Reference Framework for Software Product Management," in *Proceedings of the 14th IEEE International Requirements Engineering Conference*: IEEE Computer Society, 2006.

[3] J. Vähäniitty, "Key Decisions in Strategic New Product Development for Small Software Product Businesses," in *Proceedings of the 29th Conference on EUROMICRO*: IEEE Computer Society, 2003.

[4] T. Gorschek and C. Wohlin, "Requirements Abstraction Model," *Requirements Engineering*, vol. 11, pp. 79-101, 2006.

[5] A. Aurum and C. Wohlin, "Aligning Requirements with Business Objectives: A Framework for Requirements Engineering Decisions," in *Proceedings of Requirements Engineering Decision Support Workshop* Paris, France, 2005.

[6] T. Gorschek and A. M. Davis, "Requirements engineering: In search of the dependent variables," *Information and Software Technology*, vol. 50, pp. 67-75, 2008.

[7] F. P. Deek, J. A. M. McHugh, and O. M. Eljabiri, *Strategic Software Engineering: An Interdisciplinary Approach*: Auerbach publications, 2005.

[8] L. B. S. Raccoon, "Fifty years of progress in software engineering," *SIGSOFT Softw. Eng. Notes*, vol. 22, pp. 88-104, 1997.

[9] B. Boehm, "A view of 20th and 21st century software engineering," Shanghai, China, 2006, pp. 12-29.

[10] B. Boehm and K. Sullivan, "Software economics: Status and prospects," *Information and Software Technology*, vol. 41, pp. 937-946, 1999.

[11] B. Boehm and A. Jain, "A value-based software process framework," Shanghai, China, 2006, pp. 1-10.

[12] R. Kazman and D. Port, "Strategic Software Engineering," in *Proceedings of the 38th Hawaii International Conference on System Sciences - 2005*, 2005.

[13] A. S. Dver, *Software Product Management Essentials*. Tampa, Florida: Anclote Press, 2003.

[14] D. R. Lehmann and R. S. Winer, *Product management*, 3. ed. Boston MA: McGraw-Hill, 2002.

[15] C. M. Crawford and C. A. D. Benedetto, *New Products Management*, 7th ed.: McGraw-Hill/Irwin, 2002.

[16] N. P. Repenning, "Understanding fire fighting in new product development," *Journal of Product Innovation Management*, vol. 18, pp. 285-300, 2001.

[17] M. Crawford and A. Di Benedetto, *New Products Management*, Eight International Edition ed. Singapore: McGraw-Hill, 2006.

[18] L. Karlsson, Å. Dahlstedt, J. Natt och Dag, B. Regnell, and A. Persson, "Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study," in *Proceedings of the Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02)*, Essen, Germany, 2003, pp. 101-112.

[19] C. Wohlin, A. Aurum, and (editors), *Engineering and managing software requirements*, 1st ed. New York, NY: Springer, 2005.

[20] C. Potts, "Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software," in *Proceedings of the Second IEEE International Symposium on Requirements Engineering* Los Alamitos, 1995, pp. 128-130.

[21] B. Regnell, P. Beremark, and O. Eklundh, "A Market-Driven Requirements Engineering Process - Results from an Industrial Process Improvement Programme," *Requirements Engineering*, vol. 3, pp. 121-129, 1998.

[22] A. M. Davis, "The art of requirements triage," *IEEE Computer*, vol. 36, pp. 42-49, 2003.

[23] A. Dahlstedt, L. Karlsson, A. Persson, J. NattochDag, and B. Regnell, "Market-Driven Requirements Engineering Processes for Software Products – a Report on Current Practices," in *the International Workshop on COTS and Product Software RECOTS, held in conjunction with the 11th IEEE International Requirements Engineering Conference*, Los Alamitis CA, 2003, p. N/A.

[24] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. Natt och Dag, "An industrial survey of requirements interdependencies in software product release planning," in *Proceedings of the Fifth IEEE International Symposium on Requirements Engineering*, Los Alamitos CA, 2001, pp. 84-92.

[25] P. Carlshamre and B. Regnell, "Requirements lifecycle management and release planning in market-driven requirements engineering processes," in *Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, Los Alamitos CA, 2000, pp. 961-965.

[26] M. Höst, B. Regnell, J. Natt och Dag, J. Nedstam, and C. Nyberg, "Exploring Bottlenecks in Market-Driven Requirements Management Processes with Discrete Event Simulation," *The Journal of Systems and Software*, vol. 59, pp. 323-332, 2001.

[27] A. C. Yeh, "Requirements engineering support technique (REQUEST): a market driven requirements management process," in *Proceedings of the Second Symposium on Assessment of Quality Software Development Tools*, Los Alamitos CA, 1992, pp. 211-223.

[28] T. Gorschek and C. Wohlin, "Packaging Software Process Improvement Issues - A Method and a Case Study," *Software: Practice & Experience*, vol. 34, pp. 1311-1344, 2004.
[29] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "Industry evaluation of the Requirements Abstraction Model," *in print Requirements Engineering*.
[30] M. Khurum, K. Aslam, and T. Gorschek, "MERTS – A method for early requirements triage and selection utilizing product strategies," in *APSEC 2007* Nagoya Japan, 2007.
[31] M. Khurum and T. Gorschek, "A Method for evaluation of Alignment of Strategies among Stakeholders (MASS)," *submitted to IEEE Transactions on Engineering Management (TEM)*, 2009.
[32] R. S. Faulk, R. Harmon R., and D. Raffo M., "Value-Based Software Engineering," in *Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions* Denver, Colorado, United States: Kluwer Academic Publishers, 2000.
[33] M. Moon, K. Yeom, and H. S. Chae, "An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line," *IEEE Transactions on Software Engineering*, vol. 31, pp. 551-569, 2005.
[34] Y. P. Shin and D. K. Soo, "A systematic method for scoping core assets in product line engineering," in *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*. vol. 2005 Los Alamitos, CA 90720-1314, United States: IEEE Computer Society, 2005, pp. 491-498.
[35] K. Schmid, "A comprehensive product line scoping approach and its validation," in *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002* New York, NY, USA: ACM, 2002, pp. 593-603.
[36] K. Schmid, "The product line mapping approach to defining and structuring product portfolios," in *Proceedings IEEE Joint International Conference on Requirements Engineering* Los Alamitos, CA, USA: IEEE Comput. Soc, 2002, pp. 219-26.
[37] M. Khurum and T. Gorschek, "A systematic review of domain analysis solutions for software product lines," *Submitted to the Journal of Systems and Software*, 2009.
[38] C. Wohlin and A. Aurum, "What is important when deciding to include a software requirement in a project or release?," *Empirical Software Engineering, 2005. 2005 International Symposium on*, pp. 237-246, 2005.
[39] R. Berntsson Svensson, " Managing Quality Requirements in Software Product Development," in *Department of Computer Science*. vol. Licentiate: Lund University, 2009.
[40] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," *Information and Software Technology*, vol. 51, pp. 7-15, 2009.
[41] O. Gomez, H. Oktaba, M. Piattini, and F. Garcia, "A systematic review measurement in software engineering: state-of-the-art in measures," Setubal, Portugal, 2006, p. 8 pp.
[42] T. Dyba and T. Dingsoyr, "Empirical studies of agile software development: a systematic review," *Information and Software Technology*, vol. 50, pp. 833-59, 2008.
[43] C. Catal and B. Diri, "A systematic review of software fault prediction studies," *Expert Systems with Applications*, vol. 36, pp. 7346-7354, 2009.
[44] R. L. Glass, I. Vessey, and V. Ramesh, "Research in software engineering: An analysis of the literature," *Information and Software Technology*, vol. 44, pp. 491-506, 2002.
[45] D. E. Perry, A. A. Porter, and L. G. Votta, "Empirical studies of software engineering: a roadmap," in *Proceedings of the Conference on The Future of Software Engineering* Limerick, Ireland: ACM, 2000.

[46] E. Mendes, "A systematic review of Web engineering research," in *International Symposium on Empirical Software Engineering, 2005.*, 2005, p. 10 pp.

[47] D. Avison, J. Jones, P. Powell, and D. Wilson, "Using and validating the strategic alignment model," *Journal of Strategic Information Systems,* vol. 13, pp. 223-46, 2004.

[48] S. J. Bleistein, K. Cox, and J. Verner, "Strategic alignment in requirements analysis for organizational IT: An integrated approach," New York, NY 10036-5701, United States, 2005, pp. 1300-1307.

[49] Y. Chan, S. Huff, D. Barclay, and D. Copeland, "Business strategic orientation: information systems strategic orientation and strategic alignment," *Information Systems Research,* 1997.

[50] A. J. Hale and P. B. Cragg, "Measuring strategic alignment in small firms," Los Alamitos, CA, USA, 1996, pp. 127-35.

[51] J. C. Henderson and N. Venkatramen, "Understanding strategic alignment," *Business Quarterly,* vol. 55, pp. 72-8, 1991.

[52] J. LUFTMAN, "Assessing business-IT alignment maturity," *Communications of the Association for Information Systems,* vol. 4, 2000.

[53] J. LUFTMAN and T. BRIER, "Achieving and sustaining business-IT alignment," *California Management Review,* vol. 42, 1999.

[54] R. Papp and P. Coleman, ""Strategic alignment: analysis of perspectives," in *Southern association of Information Systems Conference,* Florida, 2006.

[55] D. Greer and G. Ruhe, "Software release planning: an evolutionary and iterative approach," *Information and Software Technology,* vol. 46, pp. 243-253, 2004.

[56] K. Joachim and R. Kevin, "A Cost-Value Approach for Prioritizing Requirements," *IEEE Softw.,* vol. 14, pp. 67-74, 1997.

[57] G, R. nther, E. Armin, and P. Dietmar, "Quantitative WinWin: a new method for decision support in requirements negotiation," in *Proceedings of the 14th international conference on Software engineering and knowledge engineering* Ischia, Italy: ACM, 2002.

[58] T. L. Saaty, *The Analytic Hierarchy Process*. New York: McGraw-Hill, 1980.

[59] G. Ruhe, A. Eberlein, and D. Pfahl, "Trade-off Analysis for Requirements Selection," *International Journal of Software Engineering and Knowledge Engineering,* pp. 345-366, 2003.

[60] M. K. Laura Lehtola, "Suitability of requirements prioritization methods for market-driven software product development," *Software Process: Improvement and Practice,* vol. 11, pp. 7-19, 2006.

[61] M. Kim, S. Park, V. Sugumaran, and H. Yang, "Managing requirements conflicts in software product lines: A goal and scenario based approach," *Data and Knowledge Engineering,* vol. 61, pp. 417-432, 2007.

[62] J. C. Dager, "Cummins's experience in developing a software product line architecture for real-time embedded diesel engine controls," in *Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions* Denver, Colorado, United States: Kluwer Academic Publishers, 2000.

[63] K. Pohl, G. Boeckle, and F. van der Linden, *Software Product Line Engineering*: Springer, 2005.

[64] M. Staples and D. Hill, "Experiences adopting software product line development without a product line architecture," in *Software Engineering Conference, 2004. 11th Asia-Pacific,* 2004, pp. 176-183.

[65] B. Gunter, M. Jesus Bermejo, K. Peter, W. K. Charles, L. Julio Cesar Sampaio do Prado, L. Frank van der, M. N. Linda, S. Michael, and M. W. David, "Adopting and Institutionalizing a Product Line Culture," in *Proceedings of the Second International Conference on Software Product Lines*: Springer-Verlag, 2002.
[66] T. Gorschek, C. Wohlin, P. Garre, and S. Larsson, "A model for technology transfer in practice," *IEEE Software,* vol. 23, pp. 88-95, Nov-Dec 2006.
[67] C. Robson, *Real world research : a resource for social scientists and practitioner-researchers*. Oxford: Blackwell, 1993.
[68] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 2nd ed.: Sage Publications, 2003.
[69] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*. Boston MA: Kluwer Academic, 2000.
[70] R. E. Stake, *The art of case study research*. Thousand Oaks: Sage Publications, 1995.
[71] J. Biolchini, P. Gomes Mian, A. Candida Cruz Natali, and G. Horta Travassos, "Systematic Review in Software Engineering," Systems Engineering and Computer Science Department, Technical Report 2005.
[72] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software,* vol. 80, pp. 571-83, 2007.
[73] B. A. Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering," in *Technical Report EBSE-2007-01*, 2007.
[74] R. K. Yin, *Case Study Research: Design and Methods*, 2nd ed.: Thousan Oaks, Sage, 1994.

Chapter 2 - A Systematic Review of Domain Analysis Solutions for Product Line

Mahvish Khurum, Tony Gorschek

Blekinge Institute of Technology, Department of Systems and Software Engineering,

S-372 25, Ronneby, Sweden

mkm@bth.se, tgo@bth.se

ABSTRACT

Domain analysis is crucial and central to software product line engineering (SPLE) as it is one of the main instruments to decide what to include in a product and how it should fit in to the overall software product line. For this reason many domain analysis solutions have been proposed both by researchers and industry practitioners. Domain analysis comprises various modeling and scoping activities. This paper presents a systematic review of all the domain analysis solutions presented until 2007. The goal of the review is to analyze the level of industrial application and/or empirical validation of the proposed solutions with the purpose of mapping maturity in terms of industrial application, as well as to what extent proposed solutions might have been evaluated in terms of usability and usefulness. The finding of this review indicates that, although many new domain analysis solutions for software product lines have been proposed over the years, the absence of qualitative and quantitative results from empirical application and/or validation makes it hard to evaluate the potential of proposed solutions with respect to their usability and/or usefulness for industry adoption. The detailed results of the systematic review can be used by individual researchers to see large gaps in research that give opportunities for future work, and from a general research perspective lessons can be learned from the absence of validation as well as from good examples presented. From an industry practitioner view, the results can be used to gauge to what extent solutions have been applied and/or validated and in what manner, both valuable as input prior to industry adoption of a domain analysis solution.

Keywords

Systematic review, domain analysis, domain modeling, domain scoping, empirical evidence, usability, usefulness

1. INTRODUCTION

Software product families have received significant attention from the software engineering community since the 1990s (Clements and Northop, 2001, DeBaud and Schmid, 1999, Deelstra et al., 2004, Dikel et al., 1997, Svahnberg and Bosch, 1999).The concept of product lines aims towards having a set of systems that share a common, managed set of features, which satisfy the particular needs of a market segment, developed from a common set of core assets in a certain given way (Clements and Northop, 2001). The product line approach is recognized as a successful approach for reuse in software development (Kim *et al.*, 2007) with the major benefits of product lines adoption reported as reduced time to market (Dager, 2000, Hetrick et al., 2006), reduced cost (Pohl et al., 2005) and improved quality (Hetrick et al., 2006, Pohl et al., 2005, Staples and Hill, 2004). For these reasons many companies developing software intensive products have either adopted or are considering the adoption of a software product line approach (Böckle, 2000, Dager, 2000).

In order to properly introduce software product lines in a company, it is important to start with the product line domain analysis. Domain analysis can be defined as "*the process by which information used in developing software systems within the domain is identified, captured, and organized with the purpose of making it reusable (to create assets) when building new products*" (America et al., 2001). This process can be used to identify commonality and variability in requirements and capture decisions on the ranges and interdependencies of variability. If domain analysis is not properly carried out, and ends up in defining either too broad or too restrictive product line scope, the major benefits like reuse, cost reduction and improved quality cannot be realized (Clements and Northop, 2001).

Several domain analysis solutions for software product lines have been presented in academia and as industry experience reports. However, in order to gauge the usability and usefulness of the proposed solutions, it is important to see the empirical evidence of their application and/or validation, e.g. in industry or through experiments or tests. Furthermore, awareness has increased in the software engineering community about the importance of empirical studies to develop or improve processes, methods and tools for software development and maintenance (Sjøberg et al., 2005). This paper presents a systematic review conducted on the studies, which either proposed or reported on experience with domain analysis solutions or parts of it (e.g. feature modeling, commonality and variability analysis, scoping and so on), presented between the years 1998 to 2007. The motivation was to gauge the level of actual industry adoption, i.e. to what extent the presented solutions are applied and/or validated in industry. In addition to industry validation, all other types of empirical results are collected to offer a detailed summation of the empirical evidence available. To achieve this, the selected studies are categorized and analyzed from several perspectives, such as research basis, application/validation method, level of validation and type of empirical results in relation to usability and usefulness of the proposed solutions. For industry practitioners looking to adopt a domain analysis solution the results of the study can be used as an indication of maturity as well as to estimate potential risk of adopting a certain solution. From an academic point of view researchers planning studies and evaluation of a solution can use this study as an inspiration for study design because the evaluation criteria of the review presented in this paper could be seen as a checklist to ascertain usability and usefulness.

The remainder of the paper is structured as follows. Section 2 describes the background and related work. Section 3 presents the research questions and design details of the review. Section 4 contains the results of the review as well as the categorization of the studies. In Section 4.2 results of data extraction and a detailed analysis are presented in relation to the research questions posed in Section 3.1. Section 5 concludes the paper.

2. BACKGROUND AND RELATED WORK

In this section the purpose of domain analysis activities are introduced. The purpose of this is twofold, one, to provide a background to the concepts relevant for this systematic review, and, two, to describe the scope of the study.

Domain analysis is the first phase of domain engineering. Domain can be defined as an area of business/technology processes or knowledge, which is described by a set of concepts and terminology understood by the stakeholders in that area (America et al., 2001). Domains are areas of expertise that can be used for creation of a system or a set of systems (Clements and Northop, 2001). The purpose of domain analysis is to gather and organize the information that is required for the smooth flow in the subsequent phases of domain engineering e.g. domain design

(Clements and Northop, 2001). Domain analysis helps in the identification of the specification of the systems in the product family. It involves various activities which can be categorized as modeling and scoping (America et al., 2001). Modeling is defined as capturing information and organizing it into a model whereas scoping is defined as a decision-making activity.

In the modeling category, the activities identified are (America et al., 2001):

1. Conceptual modeling contains a set of activities which identify, define, and organize the concepts relevant to the domain and their mutual relationships, to assist in formulating a precise and concise description of the domain. Information modeling is an important part of conceptual modeling.
2. Requirements' modeling contains a set of activities that capture the functional and architecturally relevant requirements for the product line and their inter-dependencies. This may also include mapping of specific constraints to requirements.
3. Commonality and variability modeling comprises a set of activities which identify similarities and differences between the requirements. This includes the distinction of requirements that are valid for the whole domain from those that are only valid in special cases, e.g., for a specific product variant. This activity is strongly connected to domain and feature modeling.
4. Domain modeling comprises a set of activities that specify the domains and their inter-dependencies.
5. Feature modeling comprises a set of activities which identify, study, and describe features appropriate in a given domain. The objective of feature modeling is to express relations between features, properties of features, and/or superstructures of features e.g. a commonality and variability view. One of the important purposes of feature modeling is to help structure the requirements and define the allowed variants in a product line.
6. Scenario or use-case modeling comprises a set of activities which describe and model run-time behavior of members of the system family. This not only includes the functionality of the systems and their interactions with users, but also aspects such as security, safety, reliability, and performance.

In the scoping category, we find the following activities:

1. Domain scoping is the process of identifying appropriate boundaries for a domain which is appropriate for implementing systems in the product line (Pohl et al., 2005).
2. Product line scoping is the process of systematically developing a product portfolio definition, which identifies the specific requirements and the individual products that should be part of the product line. Scope binds a product line by defining the behaviors that are "in" and the behaviors that are "out" of the product line's scope (Clements and Northop, 2001). The result of a scoping activity is a scope definition document which becomes a product line core asset. The scope definition points out the entities with which the products in the product line will interact (that is, the product line context), and it also establishes the commonality and defines the variability of the product line (Clements and Northop, 2001).
3. Asset scoping identifies the various elements that should be reusable, i.e., the specific assets that should be part of the reuse infrastructure (core assets) as opposed to being developed application specific.

A specific domain analysis solution may not mention all these activities or distinguish between them explicitly; however it is important that these activities are discussed in relation to domain analysis. Moreover, depending on the context in which a product line is being developed some of the activities might not be relevant e.g. when only very few individual domains can be distinguished, the domain modeling activity can be omitted (America et al., 2001). Domain analysis describes the characteristics of a class of systems, and not a specific system, and the scope will apply equally to existing products and products that have yet to be defined and built. Domain analysis can occur in a variety of contexts other than “start from scratch” product lines. For example, an organization may choose to apply the product line concept to only a part of the product portfolio.

A number of studies (Catal and Diri, 2009, Dyba and Dingsoyr, 2008, Glass et al., 2002, Gomez et al., 2006, Kitchenham et al., 2009, Mendes, 2005, Perry et al., 2000) have reviewed the state of empirical research in different areas e.g. computer science, software engineering, web engineering and so on. However, to the best of our knowledge no other study has been conducted with the same focus as the review presented in this paper. This review does not aim to systematically classify proposed domain analysis solutions as methods, models, tools, framework or classify the studies according to the classification and evaluation scheme suggested in (Wieringa *et al.*, 2005). The goal of this review is to analyze practical application and validation of proposed domain analysis solutions in industry to gauge their practical usability and usefulness. In addition to this other empirical evidence is also considered e.g. evidence of usability and/or usefulness demonstrated through a controlled experiment or other type of validation.

3. DESIGN

This section gives a detailed description of the review design; a definition of terms used, and discusses the validity of the study.

3.1 Research Questions and Definitions

The four research questions driving the systematic review can be viewed in Table 1.

Table 1 – Research questions and motivation.

Research Questions	Motivation
RQ1. Are solutions, proposed for domain analysis, based on needs identified in industry?	Is the solution presented based on any need/issue/problem (called *need* from here onwards) identified in industry though empirical investigation? Examples can be process assessments, case studies, participation knowledge, surveys, observations, and so on. Both direct and indirect sources will be considered, giving the presented studies the benefit of the doubt (i.e. any indication of industry basis will be considered and accepted).
RQ2. Are solutions, proposed for domain analysis, applied and/or validated in a laboratory setting or in industry?	Is the solution presented applied/validated through e.g. a controlled experiment or in industry as a part of the paper? Any validation in industry from static validation to dynamic validation will be considered, see Gorschek et al. for details (Gorschek et al., 2006).
RQ3. Are the solutions, proposed for domain analysis, usable?	If the authors of a proposed solution or experience report have applied/validated it in industry or demonstrated its application/validation through a controlled experiment, identify any indications/evidence/reports on its usability in the context of the application/validation (usability is defined below).
RQ4. Are the solutions, proposed for domain analysis, useful?	If the authors of a proposed solution or experience report have applied/validated it in industry or demonstrated its application/validation through a controlled experiment, identify any indications/evidence/reports on its usefulness in the context of the application/validation (usefulness is defined below).

The two main terms used in the research questions, namely *usability* and *usefulness*, are defined (by the authors) and exemplified below.

Figure 1 – Definition of usability and usefulness

Usability, as can be seen in Figure 1 is defined in terms of:

- Scalability of Introduction. How scalable is the proposed solution in terms of its introduction cost including e.g. training, manuals and material, tools, pilot run, and tailoring to the organization in question?

- Scalability of Use. How scalable is the proposed solution in terms of its inputs, processing time and outputs? For example, if a feature modeling approach is proposed, can it handle industry scale problems, say a hundred features, or does it solve simplified problems with simple cases or is there any indication that industry grade scalability is possible (or even considered/mentioned/discussed by the creators of the solution)?

Scalability of Introduction and Scalability of Use point to a micro quality of a solution and that is its *efficiency*. If a proposed solution is demonstrated to have any of these aspects of efficiency, the corresponding paper is counted as having some evidence of usability.

Usefulness, as can be seen in Figure 1 is defined in terms of:

- Better Alternative Investment. For example, a proposed solution (X) is better than an alternative (maybe previously used) solution (Y) e.g. with respect to usability (as defined above) and/or return on investment etc.
- Effectiveness. The effectiveness of a proposed solution in relation to achieving goals or solving the problems it was designed for. For example, solution X reduces time-to-market by 15%.

Again, a solution demonstrating either of the two aspects of usefulness is counted as having some evidence of usefulness.

3.2 Search Strategy Development

The systematic review was performed following guidelines proposed by Kitchenham in (Kitchenham, 2007). As shown in Figure 2, a three phase search strategy was devised. In Phase 1: SPLC conference proceedings from the year 2000 up to 2007 were planned to be manually searched. This was planned for several reasons. First, SPLC is the premier forum for practitioners, researchers, and educators presenting and discussing experiences, ideas, innovations, as well as challenges in the area of software product lines. SPLC also has a relatively large industry presence. Second domain analysis is a very important field and a regularly featured sub-area to software product line engineering, for the purpose of this review domain analysis solutions in relation to product lines were of primary interest. Third, since industry representation at SPLC is fairly high this includes a large amount of industry experience reports, and as one of the main features of the review is to evaluate the level of application and/or validation of the solutions, a large amount of industry experience reports was considered positive. Fourth, through the manual scanning of SPLC proceedings a number of keywords, alternate terms and synonyms were identified:

Population: software product lines, software product family

Intervention: requirements, requirements engineering, conceptual model, requirements model, commonality and variability model, domain model, feature model, scenario model, commonality analysis, variability analysis, domain evaluation, domain scope, asset scope

Comparison intervention: not applicable as the research questions are not aimed at making a comparison. The outcomes of our interest: the level of application/validation of the proposed solutions and their usability and usefulness evidence.

Outcomes: the level of application/validation of the proposed solutions and their usability and usefulness evidence

Out of scope: domain design, domain engineering and concepts related to architecture, implementation aspects.

In terms of context and experimental design, no restrictions are enforced.

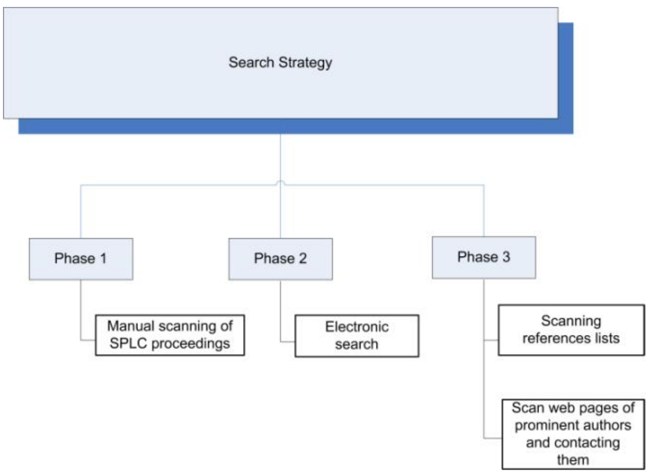

Figure 2 – 3 phase search strategy

To make the search exhaustive, in Phase 2 electronic databases were searched using the search terms deduced from the population, intervention and outcomes with the use of Boolean OR to join alternate terms and synonyms and use of Boolean AND to join major terms (Population **AND** Intervention **AND** outcomes). Examples of major terms can be seen in Appendix A. The electronic databases searched were:

1. Compendex
2. Inspec
3. IEEXplore

4. ACM digital library
5. ISI Web of Science

In order to ensure that search strings are comprehensive and precise, an expert librarian was consulted. The detail search strings are given in Appendix A.

Activities in Phase 3 were planned to ensure that any important research studies are not missed. Reference lists of the primary studies were scanned. The web pages of the authors in the particular area were also scanned.

Excluded from the search were editorials, prefaces, article summaries, interviews, news, reviews, correspondence, discussions, comments, reader's letters and summaries of tutorials, workshops, symposium, panels, and poster sessions.

3.3 Review Design

In this section, the systematic review design is presented describing studies identification method, inclusion/exclusion criteria, and the classification scheme.

3.3.1 Identification of Studies

Phase1: There were a total of 192 studies published in SPLC for the years 2000 to 2007, and in Phase 1 (see Figure 2) 24 out of 192 studies were selected after reading titles and abstracts.

Phase 2: Phase 2 had 4 steps. In Step1 (see Figure 3), 843 citations were retrieved. In Step 2 the duplicates were removed leaving 629 unduplicated citations. For all 629 citations the source of each citation, our retrieval decision, retrieval status, and eligibility decision were recorded.

In Step 3, the primary author went through all the titles to judge their relevance to the systematic review being performed. The studies whose titles were clearly not related to software product lines and domain analysis activities were excluded. For example, since our search string contained "software product line and feature model", there were studies that contained feature modeling solutions at the architecture level which were clearly out of scope, see e.g. (Zhu *et al.*, 2006). In Step 3, 359 studies were excluded leaving 270 studies in total. In Step 4, 208 studies out of 270 were excluded after reading the abstracts leaving 62 studies. The reason for excluding the 208 studies was that their focus, or main focus, was not domain analysis activities for software product lines. However, it was found that abstracts were of variable quality; some abstracts were missing, poor, and/or misleading, and several gave little indication of what was in the full article. In particular, it was not always obvious whether an experience report indeed included a domain analysis solution. If it was unclear from the title, abstract, and keywords whether a study conformed to the screening criteria, it was included giving it the benefit of doubt.

The inclusion and exclusion criteria were pilot-tested by the authors on a random sample of 15 studies. An agreement on inclusion and exclusion was achieved on 12 studies. The conflict on the remaining 3 studies was resolved after a discussion session and the inclusion/exclusion rules were refined. After the pilot, the primary author screened the remaining studies and marked them as included/excluded based on the approach described.

Phase 3: In order to ensure that no relevant studies were left out, three prominent researchers in the field of domain analysis and software product lines were consulted. As a result, three more studies were added.

This led to a selection of 89 studies in total relevant for this systematic review: 24 studies from Phase 1 + 62 studies from Phase 2 + 3 studies from Phase 3. There were no exclusions after reading the full texts.

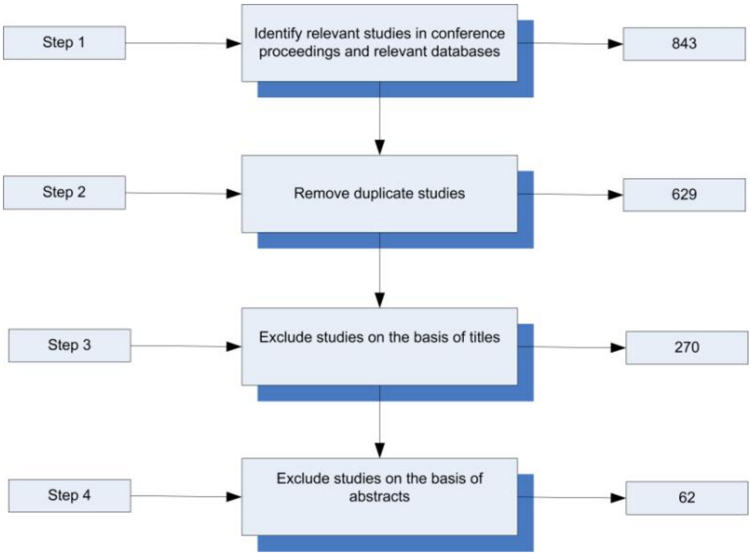

Figure 3 – Steps of Phase 2 of the search strategy

3.3.2 Quality Assessment and Data Extraction Procedure

The aim of the systematic review was to assess levels of empirical evidence and thus it did not impose any restriction in terms of any specific research method or experimental design, therefore the study quality assessment covered both quantitative and qualitative studies. The study quality assessment was primarily included in the inclusion criteria and scoping of the review, i.e. only studies that present any type of evidence or evaluation related to domain analysis for software product lines/families were included in the study. Moreover, the study quality assessment was used as a means to guide the interpretation of the findings.

Based on the research questions (see Section 3.1), a set of data extraction categories were identified with the help of guidelines from (Creswell, 2003) (Kitchenham *et al.*, 2002). Further, the categories were identified using the Goal Question Metric approach (GQM) (Basili *et al.*, 1994) during several brain storming sessions to ensure that categories identified address the aspects required to answer the research questions. Table 2 contains the definitions of the data extraction categories. There are only two categories for the research type as the purpose of the review is not to classify studies (Wieringa *et al.*, 2005) but rather find out how many and to what extent the proposed solutions are empirically applied and/or validated.

Table 2 – Definitions of the data extraction categories.

Research Type	
New Solution	Is it a new solution for domain analysis? This includes only scoping and/or modeling parts of domain analysis solution e.g. if a new method of feature modeling has been presented it is considered as a new solution.
Experience report	Is the paper an experience report describing the introduction of product lines in a company?

Empirical Basis	
Empirical	If it is “New Solution”, is it developed based on empirically identified industry needs?
Non Empirical	It is a new research idea.
Basis Reported as	If a study has empirical basis then the empirical basis reported can be categorized into 1. Statements only: the authors have written statements claiming that the need for the proposed solution has been identified in industry. 2. Participation knowledge: the authors are either practitioners in industry or participate in industry work and have identified the need for the proposed solution through participation. 3. Interviews: the authors have conducted interviews with experts in industry to identify/confirm a need and have shown that the need for the proposed solution has been identified through those interviews. 4. Process assessment: the authors have undertaken some formal process assessment e.g. using CMMI, IDEAL, REPEAT etc. and identified the need for the proposed solution.
Application/Validation	
Empirical	It is applied/validated in laboratory setting or industry.
Non Empirical	It is not applied/validated in laboratory setting or industry.
Application/Validation Method	If a study contains empirical application/validation, what method of application/validation was used? Case study, survey, interviews, experiment, observations, other, as stated by the authors. We did not differentiate between research type and research context because of the fact that almost all the studies included in the review had industry as context, thus we only differentiated between research types. If it is mentioned in the study that it has been applied/validated in industry but no description of application/validation method used is given then the method is classified as “Mentioned Industry Use”.
Application/Validation Design Explained	If a study contains empirical application/validation, the level of explanation of the design/execution of the application/validation method used is categorized into: 1. Statements: authors stated that they have applied/validated the solution in industry but no summary/details as how this was done. 2. Application/Validation summary: summary of the method without details e.g. no research questions/hypothesis, context of study, sampling of population, study execution, validity threats and so on. 3. Application/Validation in detail: a detailed explanation of the application/validation method including research questions/hypothesis, context of study, sampling of population, study execution, validity threats and so on.
Application/Validation Results Explained	If a study contains empirical application/validation, the level of explanation of the application/validation results is categorized into: 1. Nothing: no information is stated as to the results of the application/validation in the paper. 2. Statements only: the authors have written statements about the results e.g. ‘*by applying the proposed solution, time to market decreased by 15%’*. This is a statement without any results or clarification of how the results were obtained. 3. Qualitative results: for example expert opinions, e.g. 4 experts were interviewed and they foresee that application of the proposed solution would result in 15% decrease in time to market. 4. Quantitative results: collected metrics and measurements are presented. 5. Qual + Quant: when a combination of qualitative and quantitative results are presented.
Driver of Validation	If a study contains empirical application/validation, who was driving the validation of the solution in industry, was it a researcher or a practitioner? Answers can be: researcher,

	practitioner.
Replication Study	Is it a replication study? The answer could be Yes, No, Not clear, N/A (not applicable).
Builds on Paper(s)	Does the current study build on future work of some previous study published or uses and enhances any “New Solution” presented previously? This does not include a study that has been referenced in “Introduction” and/or “Related work” section. The answer could be Yes or No. If the answer to the question is yes, mention the study it is related to.
Usability & Usefulness	
Usable	Yes/No
Usability Reported as	If a study reports usability of proposed solution, the level of usability reported can be categorized into 1. Statements: the authors have written statements claiming usability e.g. “*a recent BigLever customer was able to convert their existing one-of-a-kind product into a GEARS production line with three custom product instances in less than one day*” (Krueger, 2002). 2. Qualitative data: as expert opinions e.g. 4 experts were interviewed and based on their judgment they stated the solution can be introduced in 2 days and can be applied to products with 50-500 requirements. 3. Quantitative data: e.g. “*Document examination indicated that the team understood and was able to apply all notations used after only the four hour introduction to the approach, even though they had no earlier experience of feature modeling”* (Eriksson *et al.*, 2005). 4. Qual + Quant: qualitative and quantitative data proving scalability of introduction and/or scalability of use e.g. **Quantitative data for scalability of introduction:**“*Document examination indicated that the team understood and was able to apply all notations used after only the four hour introduction to the approach, even though they had no earlier experience of feature modeling”* (Eriksson *et al.*, 2005). **Qualitative data for scalability of use:** “*Experts could not identify any scalability problems with the approach”* (Eriksson *et al.*, 2005).
Useful	Yes/No
Usefulness Reported as	If a study reports usefulness of proposed solution, the level of usefulness reported can be categorized into 1. Statements: the authors have written statements claiming usability e.g. “*It is an effective product line validation model*” (Mannion and Camara, 2003). 2. Qualitative data: as expert opinions e.g. 4 experts were interviewed and based on their judgment they stated the proposed solution will be effective than the existing method for requirements triage. 3. Quantitative data: proving effectiveness of the proposed solution e.g. data showing effective features representation and handling using proposed solution compared to existing solution. 4. Qual+Quant: qualitative and quantitative data proving effectiveness of proposed solution.
Future Work Mentioned	Has the study made promises of future work in relation to the current research? The answer can be yes/no
Written by	Is the study written by a practitioner? As stated above by practitioners it is meant the people working in industry. If the author is affiliated with industry research departments, the study is categorized as “written by practitioner”. The answer can be yes/no/not clear.

The categorization of quality attributes (usability and usefulness) into quantitative and qualitative is not intended to indicate a preference or valuation of one over the other. Any empirical data (evidence) is judged on its own merits. For example, quantitative results obtained through a controlled experiment with students as subjects might not be as valuable as the expert opinion obtained in a case study with industry practitioners who actually applied a particular solution in industry. Moreover, context, background description and design also weigh in as the purpose is to categorize the reported empirical data to analyze the levels of usability and usefulness of a proposed solution. For example, a claim about the usability and/or usefulness of the presented solution without any description of context or how the claim may be substantiated is still considered as empirical evidence from the perspective of the study, but further analysis lets the reader weigh the value of the evidence.

In order to demonstrate the mapping between research questions and the design process, Table 3 shows the research questions and the corresponding data extraction categories.

Similar to the inclusion/exclusion process, the data extraction process was tested by the authors using a sample of 10 included studies. An initial agreement on the data extraction was achieved for the data extracted from 9 studies which is quite high. For the remaining study, the consensus was reached in a discussion session. This was done to ensure that there was a common understanding of the categories defined and the classification was agreed upon by two researchers avoiding the potential bias and error source of having only one researcher performing the categorization.

Table 3 – Mapping between research questions with data extraction categories.

Research Questions	Data Extraction Categories
RQ1. Are solutions, proposed for domain analysis, based on needs identified in industry??	• Research type • Practicality – Empirical Basis • Builds on Paper(s)
RQ2. Are solutions, proposed for domain analysis, applied and/or validated in a laboratory setting or in industry?	• Research type • Practicality – Application/Validation a. Application/Validation Method b. Application/Validation Design Explained c. Application/Validation Results Explained d. Driver of Application/Validation e. Replication Study • Builds on Paper(s) • Future Work Mentioned • Written by
RQ3. Are the solutions, proposed for domain analysis, usable? RQ4. Are the solutions, proposed for domain analysis, useful?	• Usability & Usefulness a. Usability b. Usability Reported as c. Usefulness d. Usefulness Reported as

3.4 Validity Evaluation

This section presents the different validity threats related to the review and how they were addressed prior to the study to minimize the likelihood of their realization and impact.

3.4.1 Conclusion Validity

Threats to conclusion validity are related with issues that affect the ability to draw the correct conclusions from the study (Wohlin et al., 2000). From the review perspective, a potential conclusion validity threat is the reliability of the data extraction categories. To minimize this threat, GQM was used in several brain-storming sessions to extract the research questions and based on the research questions, measures (in this case the data extraction categories) were identified (see Section 3.3.2). In addition, the results presented in the review are not categorical. Any evidence, or claim made by authors are given the benefit of the doubt and counted as evidence. However, the claims are broken down and analyzed, and the value can be judged by the reader as every analysis and analysis step is transparently shown in the paper.

3.4.2 Construct Validity

Construct validity concerns generalizing the results of the study to the concept or theory behind the study (Wohlin et al., 2000). It is quite possible that the studies included in the review might not refer to the same construct using same terms thus as reviewers we might misinterpret the terms used. However, we feel fairly confident that the risk is rather minor as in addition to the term there is a context in which the term is used which minimizes the chance of misinterpretation.

From the review's perspective, another construct validity threat could be biased judgment. In this study the decision of which studies to include or exclude and how to categorize the studies could be biased and thus pose a threat. To minimize this threat both the processes of inclusion/exclusion and data extraction and coding were piloted prior to the study (see Section 3.3.1 and 3.3.2).

3.4.3 External Validity

The key idea with a systematic review is to capture as much as possible of the available literature to avoid all sorts of bias. The main challenge with a systematic review is the reliability. The reliability has been addressed as far as possible by involving two researchers, and by having a protocol which was piloted and hence evaluated. If the study is replicated by another set of researchers, it is possible that some studies that were removed in this review will be included and other studies would be excluded. However, it is highly unlikely that these random differences based on personal judgments would change the general results. It may change the actual numbers somewhat, but it is not likely that it would change the overall results as they are dominantly skewed towards one end of the spectrum (see Section 4). Thus, in general we believe that the external validity of the study is high given the use of a very systematic procedure, consultation with the researchers in the field and involvement and discussion between the two researchers.

4. RESULTS AND ANALYSIS

This section presents a summary of the results of the inclusion/exclusion procedure (see Section 4.1) as well as the analysis of data extraction from the included studies (see Section 4.2).

4.1 Included Studies Overview

Summarizing the data extracted from the included studies, 48 studies[1] out of 89 studies (both "new solution" and "experience report" types) have some form of empirical basis, all 89 studies contain some form of application/validation, and out of these 64[2] are written by researchers. The remaining 25 studies[3] are written by practitioners. None of the 89 studies is a replication study. In total 36 studies[4] out of 89 have reported on some sort of usability, and 87[5] studies out of 89 have claimed usefulness in some form.

4.2 ANALYSIS

In this section the data extracted is analyzed with respect to the research questions posed in Table 1.

4.2.1 RQ1 (Are solutions, proposed for domain analysis, based on needs identified from Industry?)

Almost half of the studies are based in some sense on the needs identified in industry (see Section 4.1). However, a deeper analysis of the empirical basis reported can be seen in Table 4 which shows that a majority of the studies have mentioned identified needs as "Statements only" (42% studies[6]), or as "Participation knowledge" (50% studies[7]). Only 2% studies[8] have mentioned interviewing experts to identify needs, and only 6% studies[9] have stated that some form of process assessment was used to identify the need for the proposed solutions.

These results make it hard to judge the credibility of the empirical basis of the solutions proposed due to the absence of presentation of e.g. process assessment and/or experts' opinions through e.g. interviews. In addition, due to the almost total lack of how the practitioners knew about the problems/needs that constitute the basis for the solutions proposed, it is impossible to draw any conclusions. In the few cases where process assessment or interviews were conducted no details such as selection criteria, method used or number of interviews, and so on are explained.

Table 4 – Included studies, "Basis" categorization.

Basis Reported as	Number of Studies	% of Studies
Statements only	20	42
Participation knowledge	24	50
Interviews	1	2
Process assessment	3	6
Total	**48**	**100**

Moreover, although a majority of the studies claims empirical basis, very few are based on future work described by previously published studies, or extend previously published solutions[10]. This may indicate that in the absence of expert interviews or proper process assessments, the needs identified may not be representative of the current problem or valid for other companies in similar situations.

The answer to RQ1 is that a majority of the proposed solutions are based on needs identified in industry, however, the actual method used and the validity of the results are impossible to ascertain as very little information is given.

4.2.2 RQ2 (Are solutions, proposed for domain analysis, applied and/or validated in a laboratory setting or in industry?)

An analysis of the studies, claiming application/validation (see Figure 4) reveals that for the years 1998 to 2007, 33% studies[11] have used case study as an application/validation method. In 24% of the studies[12] industry use was only stated and 36% studies[13] have demonstrated application/validation through simplified examples . This means that 60% of the applied/validated studies have either only mentioned industry use without any details reported, or have used simplified examples to demonstrate practicality of a proposed solution. The remaining 40% have described some details about application/validation. This makes it harder to judge the scalability of introduction and scalability of use of the proposed solutions. From Figure 4, it is evident that only 5% studies[14] have used workshops, pilots and prototyping. It is also interesting to note that for the years 1998 to 2007 only 2% studies[15] have used experimentation as a validation method.

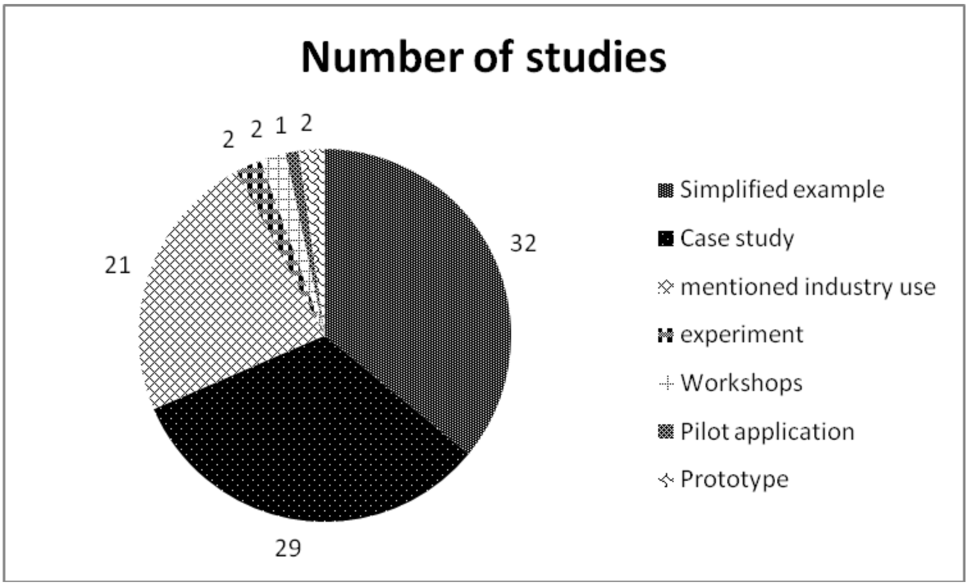

Figure 4 – Number of studies categorized according to the application/validation method

One of the reasons for these numbers could be that it is difficult to do experimentation for new solutions for product lines due to complexity of the area and difficulty in covering the entire scope in a controlled experiment. However, empirical studies are the building blocks essential for collecting evidence and to determine what situations are best for using a particular solution (Pfleeger, 1999). The current situation means there is a lack of quantitative and/or qualitative data that a new solution is better than an already existing one, or what the impact of implementing the new solution might be. This makes it impossible to gauge efficiency or effectiveness of proposed solutions either alone or in relation to better alternative investment (BAI).

Table 5 – Included studies, “Application/Validation Design Explained” categorized.

Application/Validation Design Explained	Number of Studies	% of Studies
Statements only	14	16
Application/Validation summary	45	50
Application/Validation in detail	30	34
Total	**89**	**100**

Moving on from the analysis of application/validation methods used to the analysis of the application/validation design details, Table 5 shows the categorization of the application/validation design explanation given in the included studies claiming some form of application/validation. From Table 5, it is possible to see that 84% studies either provide application/validation summary (50% studies[16]) or explain application/validation in detail (34% studies[17]). This seems to be a positive outcome that most of the studies have explained application/validation in detail. However, after analyzing the level of application/validation results, it is found that a majority of the studies either say nothing about the application/validation results (11%) or have only statements about the results (70% studies) (see first two rows of Table 6). Only 7% studies[18] provide qualitative results as experts’ opinion, 9% studies[19] provide quantitative results and only 3% studies[20] provide both qualitative and quantitative results.

Table 6 – Included studies “Application /Validation Results” categorized.

Application/Validation Results Explained	Number of Studies	% of Studies
Nothing	10	11
Statements only	63	70
Results as experts opinion	6	7
Quantitative data	7	9
Qual +Quant	3	3
Total	**89**	**100**

Thus, Table 6 reveals that out of 89 included studies with the claim of empirical evidence, 80% studies lack qualitative or quantitative results of application/validation. The absence of strong application/validation results may be one of the reasons that few studies have used previously proposed solutions (see Section 4.2.1).

Majority of the “Experience report” studies state the results of the experiences as lessons learned without any indication how these lessons were collected. The lack of description in relation to the experiences, for example if interviews were used, if there were any quantitative measures and so on, makes it difficult to judge validity. This also makes it hard for other practitioners to gauge the context and relevance of the experiences reported.

Figure 5 presents another aspect to answer this question and that is to see how many solutions from each year are based on solutions presented in previous years. Figure 5 shows that many new solutions have been presented over the years, but very few actually have been used as a basis for further development or adoption, piloting or test in industry. For example, by the year 2003 a total of 28 new solutions had been proposed but only 5 studies reported the use of any of the previously proposed solutions (in industry or as a basis for refinement of a solution). By 2007 the number of "New Solution" studies had reached 73, and only 12 studies were based on previously proposed solutions or reported experience based on the use of previously proposed solutions. This may indicate that the proposed solutions are not applicable in industry or that due to missing application/validation results the solutions are not applied by practitioners and not used by researchers. This problem has been indicated by others as well e.g. in (Kircher *et al.*, 2006). This may imply that a focus on validation and proper reporting should be premiered over the continuous presentation of new solutions. Another possibility is that the proposed solutions do not solve the challenges in industry, which in turn implies that there is a need to understand the challenges. Another possible conclusion could be that industry practitioners are not up to date with the new solutions proposed, thus the solutions go unused. None of the studies presented from the year 1998 to 2007 were replicated studies.

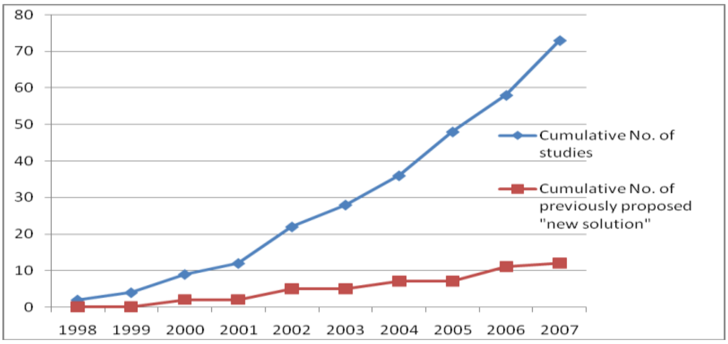

Figure 5 – Cum No. of studies VS Cum No. of previously proposed "new solution" studies

Summarizing all the aspects analyzed above, the answer to RQ2 is that although the studies from the years 1998 to 2007 have reported some form of application/validation, the absence of detailed results or replicated studies make it difficult to evaluate the potential of the proposed solutions. This is further compounded by the fact that the validity of the reported experiences reports is also very hard to judge.

4.2.3 Answering RQ3 (Are the solutions, proposed for domain analysis, usable?) and RQ4 (Are the solutions, proposed for domain analysis, useful?)

In 36 studies usability was mentioned as a part of the proposed solutions. However, looking at Table 7 it is possible to see that 80% of the studies[21] only have statements claiming usability. An example of this can be illustrated by the following statement*: "A minor problem occurs as the table can grow and become unwieldy for large application areas, but this can be addressed by segmenting the table appropriately*" (DeBaud and Schmid, 1999). In 8% of the studies[22] qualitative evidence of usability as expert opinion was presented, for example, "*After finishing the project, the project manager and developers agreed that the proposed domain requirements development approach was very helpful for identifying and specifying application requirements, resulting in reducing the overall development effort*" (Moon *et al.*, 2005). 8% of the studies[23] gave quantitative evidence of usability and only 2% of the studies[24] gave both qualitative and quantitative evidence of usability e.g. "*…indicated that the team understood and was able to apply all notations used after only the four hour introduction to the approach, even though they had no earlier experience of feature modeling*", results of "*questionnaire indicated that the* *product line analysis team gained a better understanding of the domain during the modeling activity*" (Eriksson *et al.*, 2005).

Table 7 – "Usability" reported in included studies.

Usability	Number of studies	% of Studies
Statements only	29	80
Results as experts opinion	3	8
Quantitative results	3	8
Qual +Quant	1	2
Total	**36**	**100**

Clearly with the usability statements as exemplified above, it is difficult to judge the usability of a proposed solution. A clear majority of the studies do not include either qualitative or quantitative data about scalability of introduction or scalability of use, making it harder for practitioners to evaluate usability. It is important to understand that the intention in this review is not to criticize studies but to highlight the absence of qualitative and quantitative evidence in relation to usability, which might be a barrier for the industrial adoption of the proposed solutions.

Table 8 - "Usefulness" reported in included studies.

Usefulness	Number of Studies	% of Studies
Statements only	70	81
Results as experts opinion	9	10
Quantitative results	6	7
Qual +Quant	2	2
Total	**87**	**100**

Positive results regarding usefulness were reported by 87 of the included studies, which seem to be very good, but a deeper analysis as can be seen in Table 8 shows that 81% of the studies[25] claim usefulness as statements. For example, "*The ILP modeling approach presented in the former section was tested in a Stago project with satisfying results*" (Djebbi and Salinesi, 2007). There are only 2% of the studies[26] that provides qualitative and quantitative data about the usefulness of the proposed solution.

If the percentages and the categorization of reported application/validation results are kept in mind, it seems logical that since the application/validation results were mostly statements (see Table 6), usability and usefulness evidence would naturally also be statements due to the absence of qualitative and/or quantitative results for the application/validation of a solution. This also results in difficulty to find any qualitative or quantitative evidence of usability or usefulness, even in the form of statements and claims, made by the authors of respective studies.

The answer to RQ3 and RQ4 is that although there are statements regarding usability and usefulness in the studies published for the years 1998 to 2007, lack of qualitative and quantitative data of any sort makes it difficult to evaluate how usable and useful the proposed solutions/experiences are. In industry, time and resources are scarce. If a practitioner cannot clearly determine the time and resources required to implement a solution against the usefulness of the solution in comparison to available better alternative investments, it is very unlikely that the solution will be adopted based only on statements made by the creators of the solution. Similarly, if authors do not show scalability of use of a particular solution indication the ability to tackle industry scale problems, practitioners would probably not take the risk of implementing a solution, as it falls short on reporting even rudimentary evidence on efficiency.

5. CONCLUSION

This paper presents the systematic review of the modeling and scoping activities involved in domain analysis for software product lines from the year 1998 until 2007. With a three phase search strategy, 89 studies were selected that either proposed new solutions of domain analysis or reports of experiences in using such solutions. In order to *analyze the practical application and validation of proposed domain analysis solutions in industry and to gauge their practical usability and usefulness,* four research questions were specified (see Section 3.1.). Based on the goal and corresponding research questions to achieve that goal, a data extraction procedure was defined (see Section 3.3.2.). Data was then extracted using a defined procedure covering the basis of a study, practicality, usability and usefulness, future work and information about the authors.

The major findings of the review can be summarized as follows:

1) Many domain analysis solutions have been presented over the years and a majority of the studies address needs identified in industry, but they fall short on the approach used to identify the need for a solution. Most studies only claim that they based the solution on a need identified in industry or state that through participation knowledge the need for the proposed solution was identified. Such claims and statements may be valid, but they raise validity questions both from a research perspective and an industrial adoption perspective. Without interviewing experts in industry or performing some form of process assessment, it is hard to triangulate the need identified thus raising the issue that the need may not be representative of the current situation. As a result, this poses questions about the internal and external validity of the needs identified, and this is passed on to the corresponding solutions proposed.

2) Many studies claim that they have applied/validated the proposed solutions in industry; however, a deeper analysis reveals that a majority of the claims are merely statements (80%), and qualitative and quantitative evidence supporting these claims is generally missing. Claims and statements may be valid, but in the absence of clear qualitative evidence as experts' opinions and/or quantitative data about the benefits of the proposed solution, it is hard to evaluate the potential of these solutions for industry adoption.

3) Many studies claim usability and usefulness of the proposed solutions in some form, however a deeper analysis reveals that majority of the claims are also merely statements about usability (80%) and usefulness (81%). As mentioned previously such claims may be valid, but they raise validity questions from both a research and industrial adoption perspective. Without experts' opinions and/or quantitative data supporting the usability and usefulness claims, it is difficult to evaluate the validity of the claims, and similarly it is difficult for the

practitioners to evaluate the usability and usefulness of a proposed solution for application in industry.

The overall goal of this review was not to expect or demand perfect evidence of usability and usefulness following perfect and extensive data collection in industry. However, many studies over the years have shown that it is possible to validate proposed solutions in any number of ways. Controlled experiments could be used in academia, even if the use of students as subjects is debated. Traditionally, experiments in software engineering were performed on a limited scale e.g. comparing defects detection techniques (Lott and Rombach, 1996). In the context of software product lines specific techniques can also be tested e.g. comparing different feature modeling techniques however, testing areas such as scoping and requirements engineering decisions in SPL are harder to simulate in a controlled environment. One of the contributions of this paper is to highlight the fact that refined experiment designs might be needed keeping in view the inherent complexity and broader scope of software product lines. Static (preliminary) validation may be performed in industry through workshops, interviews, or surveys. Dynamic validation (e.g. pilots) may be performed collecting metrics and qualitative data though interviews with practitioners. The data collected is not complete, but vastly better than no data at all.

In addition to doing validation (e.g. in industry), the way in which the validation is planned and reported is also crucial. The studies reviewed are full of statements, claiming usability and usefulness. The good thing is that this indicates that our interest in these two concepts in this systematic review is relevant, i.e. usability and usefulness of solutions are important and this is confirmed by the authors themselves. However, even if statements are common, very little evidence is presented, both in terms of absence of data, but also absence of design for the studies presented. The only seemingly complete validation is when there is no real validation, e.g. in case of presenting simplified examples. The use of simplified examples is not without merit, e.g. it may be used to explain and exemplify the use of a solution initially, but use of a simplified example is not the same as validation, even if the example is based on something relevant for industry. One might even go so far as to expect an evolution, that is, a new solution proposed is exemplified and explained through the use of simplified and scaled down examples in initial publications, then validation is performed, scaling up the tests of the solution.

The presence of empirical evidence of any sort with at least some intent to explain the overall design and execution of a validation (e.g. a pilot test in industry) could be very beneficial for both researchers and industry practitioners. From an academic point of view the possibility to learn and extend on presented research is crucial for progress. In addition, one of the foundations of research is the possibility to replicate studies. None of the studies included was a replicated study.

From a practitioner point of view, a design and illustration of how conclusions about usability and usefulness are made can vastly improve the relevance of any paper. The total absence of data or evidence is problematic from two perspectives. First, can the results be trusted? Second, even if the authors are given the benefit of the doubt, is the proposed solution relevant for all cases? If not, what cases?

There may be several explanations for the results of this systematic review. One could be that the included conference and journals attract a certain type of studies that do not focus on empirical results. Another explanation could be that in case of conferences a ten page limit presents problems for presenting empirical results, even if there are many studies who manage (some examples from SPLC conference are (Eriksson et al., 2005) and (Jepsen et al., 2007, Lee et al., 2000)). Moreover, guidelines for conducting empirical research has been presented in a number of papers e.g. (Jedlitschka and Pfahl, 2005, Kitchenham et al., 2008, Kitchenham et al., 2002, Runeson and Höst, 2009, Staples and Niazi, 2007) which can be used. Yet another explanation could be that industry validation is hard to achieve. The question is, should we accept these explanations, or should we strive for improving state-of-the-art reporting?

Summarizing the contribution of this systematic review we have two main perspectives. For industry practitioners looking to adopt a domain analysis solution, the results of the study can be used as an indication of maturity as well as to estimate potential risk of a certain solution before considering its application. From an academic point of view researchers planning studies and evaluation of a solution can use this study as inspiration for study design as the evaluation criteria of the review presented in this paper could be seen as a checklist to ascertain usability and usefulness.

6. Acknowledgements

We will like to thank Klaus Schmid, Lianping Chen, Mikael Svahnberg and Kent Pettersson who helped us in binding the scope of this systematic review and making searches as exhaustive as possible.

7. REFERENCES

America, P., Thiel, S., Ferber, S. and Mergel, M., 2001, Introduction to Domain Analysis,

Basili, V., Caldiera, G. and Rombach, H. D., 1994, The Goal Question Metric Approach, John Wiley & Sons,

Böckle, G., 2000, Model-based requirements engineering for product lines, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers

Catal, C. and Diri, B., 2009, A systematic review of software fault prediction studies, Expert Systems with Applications, 36, 4, 7346-7354

Clements, P. and Northop, L., 2001, Software Product Lines: Practices and Patterns, Addisson-Wesley, 0-201-70332-7

Creswell, J. W., 2003, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, Sage Publications,

Dager, J. C., 2000, Cummins's experience in developing a software product line architecture for real-time embedded diesel engine controls, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers

DeBaud, J. M. and Schmid, K., 1999, A systematic approach to derive the scope of software product lines, Proceedings of the 21st international conference on Software,

Deelstra, S., Sinnema, M. and Bosch, J., 2004, Experiences in Software Product Families: Problems and Issues During Product Derivation, Proceedings of Third International Conference, SPLC2004, Boston, MA, USA, Springer

Dikel, D., Kane, D., Ornburn, S., Loftus, W. and Wilson, J., 1997, Applying software product-line architecture, Computer, 30, 8, 49-55

Djebbi, O. and Salinesi, C., 2007, RED-PL, a method for deriving product requirements from a product line requirements model, Springer Verlag, Heidelberg, D-69121, Germany

Dyba, T. and Dingsoyr, T., 2008, Empirical studies of agile software development: a systematic review, Information and Software Technology, 50, 9-10, 833-59

Eriksson, M., Börstler, J. and Borg , K., 2005, The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations, Ninth International Conference on Software Product Lines, Rennes, France,

Glass, R. L., Vessey, I. and Ramesh, V., 2002, Research in software engineering: An analysis of the literature, Information and Software Technology, 44, 8, 491-506

Gomez, O., Oktaba, H., Piattini, M. and Garcia, F., 2006, A systematic review measurement in software engineering: state-of-the-art in measures, Institute for Systems and Technologies of Information, Control and Communication
Gorschek, T., Wohlin, C., Garre, P. and Larsson, S., 2006, A model for technology transfer in practice, IEEE Software, 23, 6, 88-95
Hetrick, W. A., Krueger, C. W. and Moore, J. G., 2006, Incremental return on incremental investment: Engenio's transition to software product line practice, Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications, Portland, Oregon, USA, ACM
Jedlitschka, A. and Pfahl, D., 2005, Reporting guidelines for controlled experiments in software engineering, Institute of Electrical and Electronics Engineers Computer Society
Jepsen, H. P., Dall, J. G. and Beuche, D., 2007, Minimally Invasive Migration to Software Product Lines, Proceedings of the 11th International Software Product Line Conference, IEEE Computer Society
Kim, M., Park, S., Sugumaran, V. and Yang, H., 2007, Managing requirements conflicts in software product lines: A goal and scenario based approach, Data and Knowledge Engineering, 61, 3, 417-432
Kircher, M., Schwanninger, C. and Groher, I., 2006, Transitioning to a Software Product Family Approach - Challenges and Best Practices, Proceedings of the 10th International on Software Product Line Conference, IEEE Computer Society
Kitchenham, B., Al-Khilidar, H., Babar, M. A., Berry, M., Cox, K., Keung, J., Kurniawati, F., Staples, M., Zhang, H. and Zhu, L., 2008, Evaluating guidelines for reporting empirical software engineering studies, Empirical Software Engineering, 13, 1, 97-121
Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J. and Linkman, S., 2009, Systematic literature reviews in software engineering - A systematic literature review, Information and Software Technology, 51, 1, 7-15
Kitchenham, B. A., 2007, Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE-2007-01,
Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K. and Rosenberg, J., 2002, Preliminary guidelines for empirical research in software engineering, IEEE Transactions on Software Engineering, 28, 8, 721-34
Krueger, C. W., 2002, Easing the Transition to Software Mass Customization, Revised Papers from the 4th International Workshop on Software Product-Family Engineering, Springer-Verlag
Lee, K., Kang, K. C., Koh, E., Chae, W., Kim, B. and Choi, B. W., 2000, Domain-oriented engineering of elevator control software: a product line practice, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers
Lott, C. M. and Rombach, H. D., 1996, Repeatable software engineering experiments for comparing defect-detection techniques, Empirical Software Engineering, 1, 3, 241-77
Mannion, M. and Camara, J., 2003, Theorem proving for product line model verification, 5th International Workshop Software Product - Family Engineering., Siena, Italy,
Mendes, E., 2005, A systematic review of Web engineering research, International Symposium on Empirical Software Engineering, 2005.,
Moon, M., Yeom, K. and Chae, H. S., 2005, An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line, IEEE Transactions on Software Engineering, 31, 7, 551-569
Perry, D. E., Porter, A. A. and Votta, L. G., 2000, Empirical studies of software engineering: a roadmap, Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland, ACM
Pfleeger, S. L., 1999, Albert Einstein and empirical software engineering, Computer, 32, 10, 32-38
Pohl, K., Boeckle, G. and van der Linden, F., 2005, Software Product Line Engineering, Springer, 10 3-540-24372-0 Springer Berlin Heidelberg New York
Runeson, P. and Höst, M., 2009, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering, 14, 2, 131-164
Sjøberg, D. I. K., Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N. K. and Rekdal, A. C., 2005, A survey of controlled experiments in software engineering, Software Engineering, IEEE Transactions on, 31, 9, 733-753

Staples, M. and Hill, D., 2004, Experiences adopting software product line development without a product line architecture,
Staples, M. and Niazi, M., 2007, Experiences using systematic review guidelines, Journal of Systems and Software, 80, 9, 1425-1437
Svahnberg, M. and Bosch, J., 1999, Evolution in software product lines: two cases, Journal of Software Maintenance: Research and Practice, 11, 6, 391-422
Wieringa, R., Maiden, N., Mead, N. and Rolland, C., 2005, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requir. Eng., 11, 1, 102-107
Wohlin, C., Runeson, P., Höst, M., Ohlson, M. C., Regnell, B. and Wesslén, A., 2000, Experimentation in Software Engineering: An Introduction, Kluwer Academic, 0792386825 (acid-free paper)
Zhu, C., Lee, Y., Zhao, W. and Jingzhou, Z., 2006, A feature oriented approach to mapping from domain requirements to product line architecture, Proceedings of the 2006 International Conference on Software Engineering Research and Practice and Conference on Programming Languages and Compilers SERP'06, USA, CSREA Press

Appendix A – Search Strings

Search Strings	Databases
(((("product famil*" OR "product line*") AND ({requirements} OR {requirements engineering*} OR {conceptual model*} OR {requirements model*} OR {commonality and variability model*} OR {domain model*} OR {feature model*} OR {scenario model*} OR {commonality analysis} OR {variability analysis} OR {domain eval*} OR {domain scop*} OR "asset scop*")) AND ((((("product famil*" OR "product line*") AND ({requirements} OR {requirements engineering*} OR {conceptual model*} OR {requirements model*} OR {commonality and variability model*} OR {domain model*} OR {feature model*} OR {scenario model*} OR {commonality analysis} OR {variability analysis} OR {domain eval*} OR {domain scop*} OR "asset scop*")) AND (empiric* OR experience* OR "lesson learned" OR "lesson learnt" OR "lessons learned" OR evaluat* OR validat* OR experiment* OR stud* OR case* OR example* OR survey OR analys* OR investig* OR demonstrate* OR industr*) WN KY) AND (1969-2007 WN YR) AND (English) WN LA)) WN KY) AND (1969-2007 WN YR) AND (English) WN LA)	Inspec and Compendex via Engineering Village2
(((Abstract:product and Abstract:line) OR (Abstract:product and Abstract:famil*)) AND ((Abstract:requirements and Abstract: Model*) OR (Abstract:requirements and Abstract: engineer*) OR (Abstract:requirements) OR (Abstract:conceptual and Abstract:Model*) OR (Abstract:feature and Abstract: Model*) OR (Abstract:commonality and Abstract:analysis) OR (Abstract:variability and Abstract:analysis) OR (Abstract:domain and Abstract:scop*) OR (Abstract:domain and Abstract:eval*) OR (Abstract:Asset and Abstract:scop*)) AND ((Abstract:case and Abstract:stud*) OR (Abstract:empiric*) OR (Abstract:experienc*) OR (Abstract:lessons and Abstract:learn*) OR (Abstract:evaluate*) OR (Abstract:validate*) OR (Abstract:experiment*) OR (Abstract:example*) OR (Abstract:survey*) OR (Abstract:analy*) OR (Abstract:investigat*) OR (Abstract:demonstrat*) OR (Abstract:industr*)))	ACM
((((product line)<in>ab) <or> ((product famil*)<in>ab)) <and> (((requirements model*)<in>ab) <or> ((requirements engineer*)<in>ab) <or> ((requirements)<in>ab) <or> ((conceptual model*)<in>ab) <or> ((feature model*)<in>ab) <or> ((commonality analysis)<in>ab) <or> ((variability analysis)<in>ab) <or> ((domain sco*)<in>ab) <or> ((domain eval*)<in>ab) <or> ((asset sco*)<in>ab)) <and> (((experienc*)<in>ab) <or> ((empiric*)<in>ab)<or> ((lessons learn**)<in>ab)<or> ((evaluat*)<in>ab)<or> ((validat*)<in>ab)<or> ((expeiment*)<in>ab)<or> ((case study)<in>ab)<or> ((survey*)<in>ab)<or> ((analy*)<in>ab)<or> ((investigat*)<in>ab)<or> ((demonstrat*)<in>ab)<or> ((industr*)<in>ab)))	IEEEXplore
TS=("product line" OR "product famil*" AND (TS=("requirements" OR "requirements engineering" OR "requirements model*" OR "feature model*" OR "commonality analysis" OR "variability analysis" OR "domain scop*" OR "domain eval*" OR "asset scop*")) AND (TS=("case stud*" OR "empiric*" OR "experien" OR "Lessons learn*" OR "evaluat*" or "validat*" OR "experiment" OR "exampl" OR "survey" OR "Analy*" OR "investigat*" OR "validat*" OR "industri*"))) AND Language=(English) AND Document Type=(Article)	ISI Web of Sceince

Appendix B – Selected Studies

Study Id	Study Name
1	Jarzabek, S., Yang, B. and Yoeun, S., 2006, Addressing quality attributes in domain analysis for product lines, IEE Proceedings: Software, 153, 2, 61-73
2	Thompson, J. M. and Heimdahl, M. P. E., 2003, Structuring product family requirements for n-dimensional and hierarchical product lines, Requirements Engineering, 8, 1, 42-54
3	Knauber, P., Muthig, D., Schmid, K. and Widen, T., 2000, Applying product line concepts in small and medium-sized companies, Ieee Software, 17, 5,
4	Soon-Bok, L., Jin-Woo, K., Chee-Yang, S. and Doo-Kwon, B., 2007, An approach to analyzing commonality and variability of features using ontology in a software product line engineering, 5th International Conference on Software Engineering Research, Management and Applications, Piscataway, NJ, USA, IEEE
5	Kun, C., Wei, Z., Haiyan, Z. and Hong, M., 2005, An approach to constructing feature models based on requirements clustering, Proceedings of 13th IEEE International Conference on Requirements Engineering, Los Alamitos, CA, USA, IEEE Comput. Soc
6	Yuqin, L., Chuanyao, Y., Chongxiang, Z. and Wenyun, Z., 2006, An approach to managing feature dependencies for product releasing in software product lines, 9th International Conference on Software Reuse, ICSR 2006. Proceedings (Lecture Notes in Computer Science Vol.4039), Berlin, Germany, Springer-Verlag
7	Kuloor, C. and Eberlein, A., 2003, Aspect-oriented requirements engineering for software product lines, Proceedings 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, Los Alamitos, CA, USA, IEEE Comput. Soc
8	Lutz, R. R. and Qian, F., 2005, Bi-directional safety analysis of product lines, Journal of Systems and Software, 78, 2, 111-27
9	Lam, W., 1998, A case-study of requirements reuse through product families, Annals of Software Engineering, 5, 253-77
10	Ramachandran, M. and Allen, P., 2005, Commonality and variability analysis in industrial practice for product line improvement, Software Process Improvement and Practice, 10, 1, 31-40
11	Schmid, K., 2002a, A comprehensive product line scoping approach and its validation, Proceedings of the 24th International Conference on Software Engineering. ICSE 2002, New York, NY, USA, ACM
12	DeBaud, J.-M. and Schmid, K., 1999, A systematic approach to derive the scope of software product lines, Proceedings of the 21st international conference on Software engineering, Los Angeles, California, United States, ACM
13	Bayer, J., Flege, O., Knauber, P., Laqua, R., Muthig, D., Schmid, K., Widen, T. and DeBaud, J. M., 1999, PuLSE: A methodology to develop software product lines, SSR'99. Proceedings of the Fifth Symposium on Software Reusability. Bridging the Gap Between Research and Practice, New York, NY, USA, ACM
14	Sinnema, M., Deelstra, S., Nijhuis, J. and Bosch, J., 2004, COVAMOF: a framework for modeling variability in software product families, Third International Conference on Software Product Lines, SPLC 2004. Proceedings (Lecture Notes in Comput. Sci. Vol.3154), Berlin, Germany, Springer-Verlag
15	Aubrey, D., 2006, Controlling the HMS Program through managing requirements,

	Proceedings of the IEEE International Conference on Requirements Engineering, Piscataway, NJ 08855-1331, United States, Institute of Electrical and Electronics Engineers Computer Society
16	Schmid, K. and John, I., 2004, A customizable approach to full lifecycle variability management, Science of Computer Programming, 53, 3, 259-284
17	Eisenbarth, T., Koschke, R. and Simon, D., 2001, Derivation of feature component maps by means of concept analysis, Fifth European Conference on Software Maintenance and Reengineering, 2001,
18	Djebbi, O., Salinesi, C. and Diaz, D., 2007, Deriving product line requirements: The RED-PL guidance approach, Proceedings of Asia-Pacific Software Engineering Conference, APSEC, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
19	Deelstra, S., Sinnema, M., Nijhuis, J. and Bosch, J., 2004, COSVAM: A technique for assessing software variability in software product families, IEEE International Conference on Software Maintenance, ICSM, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
20	Metzger, A., Heymans, P., Pohl, K., Schobbens, P. Y. and Saval, G., 2007, Disambiguating the documentation of variability in software product lines: a separation of concerns, formalization and automated analysis, 2007 IEEE International Conference on Requirements Engineering, Piscataway, NJ, USA, IEEE
21	Minseong, K., Hwasil, Y. and Sooyong, P., 2003, A domain analysis method for software product lines based on scenarios, goals and features, 10 Asia-Pacific Software Engineering Conference, Los Alamitos, CA, USA, IEEE Comput. Soc
22	John, I., Muthig, D., Sody, P. and Tolzmann, E., 2002, Efficient and systematic software evolution through domain analysis, Proceedings of IEEE Joint International Conference on Requirements Engineering,
23	Thurimella, A. K. and Bruegge, B., 2007, Evolution in product line requirements engineering: a rationale management approach, 2007 IEEE International Conference on Requirements Engineering, Piscataway, NJ, USA, IEEE
24	Breen, M., 2005, Experience of using a lightweight formal specification method for a commercial embedded system product line, Requirements Engineering, 10, 2, 161-72
25	Zuo, H., Mannion, M., Sellier, D. and Foley, R., 2005, An extension of problem frame notation for software product lines, Proceedings of Asia-Pacific Software Engineering Conference, APSEC, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
26	Jongsu, B. and Sungwon, K., 2007, A Method to Generate a Feature Model from a Business Process Model for Business Applications, 7th IEEE International Conference on Computer and Information Technology, 2007. CIT 2007.,
27	Kang, K. C., Jaejoon, L. and Donohoe, P., 2002, Feature-oriented product line engineering, IEEE Software, 19, 4, 58-65
28	Thiel, S. and Hein, A., 2002, Modeling and using product line variability in automative systems, IEEE Software, 19, 4, 66-72
29	Kim, J., Kim, M., Yang, H. and Park, S., 2004, A method and tool support for variant requirements analysis: Goal and scenario based approach, Proceedings of Asia-Pacific Software Engineering Conference, APSEC, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
30	Moon, M., Yeom, K. and Chae, H. S., 2005, An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product

	line, IEEE Transactions on Software Engineering, 31, 7, 551-569
31	Zhang, H. and Jarzabek, S., 2004, XVCL: A mechanism for handling variants in software product lines, Science of Computer Programming, 53, 3, 381-407
32	Savolainen, J. and Kuusela, J., 2001, Volatility analysis framework for product lines, Proceedings of SSR'01 2001 Symposium on Software Reusability, Association for Computing Machinery
33	Jeong, A. K., 2006, Variability management with ACM (Adaptable Component Model) for insurance product line, Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration, IRI-2006, Piscataway, NJ 08855-1331, United States, Institute of Electrical and Electronics Engineers Computer Society
34	Prasanna, P. and Lutz, R. R., 2005, Tool-supported verification of product line requirements, Automated Software Engineering, 12, 4, 447-65
35	Nakanishi, T., Fujita, M., Yamazaki, S., Yamashita, N. and Ashihara, S., 2007, Tailoring the Domain Engineering Process of the PLUS Method, 14th Asia-Pacific Software Engineering Conference, 2007. APSEC 2007. ,
36	Shin, Y. P. and Soo, D. K., 2005, A systematic method for scoping core assets in product line engineering, Proceedings - Asia-Pacific Software Engineering Conference, APSEC, Los Alamitos, CA 90720-1314, United States, IEEE Computer Society
37	Ardis, M., Daley, N., Hoffman, D., Siy, H. and Weiss, D., 2000a, Software product lines: a case study, Software-Practice & Experience, 30, 7, 825-847
38	Spanoudakis, G., Zisman, A., Perez-Minana, E. and Krause, P., 2004, Rule-based generation of requirements traceability relations, Journal of Systems and Software, 72, 2, 105-127
39	Yiyuan, L., Jianwei, Y., Dongcai, S., Ying, L. and Jinxiang, D., 2007, Software product line oriented feature map, Proceedings of 7th International Conference of Computational Science-ICCS 2007, Part II (Lecture Notes in Computer Science Vol.4488), Berlin, Germany, Springer
40	Eriksson, M., Borstler, J. and Borg, K., 2006, Software product line modeling made practical, Communications of the ACM, 49, 12, 1183265
41	Gomaa, H. and Saleh, M., 2006, Software product line engineering and dynamic customization of a radio frequency management system, Institute of Electrical and Electronics Engineers Computer Society
42	Gomaa, H. and Shin, M. E., 2002, Multiple-view meta-modeling of software product lines, Proceedings Eighth IEEE International Conference on Engineering of Complex Computer Systems, Los Alamitos, CA, USA, IEEE Comput. Soc
43	Laguna, M. A., Gonzalez-Baixauli, B. and Marques, J. M., 2007, Seamless development of software product lines, GPCE'07 - Proceedings of the Sixth International Conference on Generative Programming and Component Engineering, New York, NY 10036-5701, United States, Association for Computing Machinery
44	Sooyong, P., Minseong, K. and Sugumaran, V., 2004, A scenario, goal and feature-oriented domain analysis approach for developing software product lines, Industrial Management + Data Systems, 104, 4, 296-308
45	Liu, J., Dehlinger, J. and Lutz, R., 2007, Safety analysis of software product lines using state-based modeling, Journal of Systems and Software, 80, 11, 1879-1892
46	Lutz, R. R., Helmer, G. G., Moseman, M. M., Statezni, D. E. and Tockey, S. R., 1998, Safety analysis of requirements for a product family, Requirements Engineering, 1998.

	Proceedings. 1998 Third International Conference on,
47	Loesch, F. and Ploedereder, E., 2007, Restructuring variability in software product lines using concept analysis of product configurations, 11th European Conference on Software Maintenance and Reengineering, Los Alamitos, CA, USA, IEEE Computer Society
48	Kuusela, J. and Savolainen, J., 2000, Requirements engineering for product families, Proceedings of International Conference on Software Engineering, Los Alamitos, CA, USA, Institute of Electrical and Electronics Engineers Computer Society
49	Djebbi, O. and Salinesi, C., 2007, RED-PL, a method for deriving product requirements from a product line requirements model, . Proceedings 19th International Conference on Advanced Information Systems Engineering, CAiSE 2007, Berlin, Germany, Springer-Verlag
50	Clements, P. C., Jones, L. G., Northrop, L. M. and McGregor, J. D., 2005, Project management in a software product line organization, IEEE Software, 22, 5, 54-62
51	Faulk, S. R., 2001, Product-Line Requirements Specification (PRS): An approach and case study, Proceedings of the IEEE International Conference on Requirements Engineering, Institute of Electrical and Electronics Engineers Computer Society
52	Schmid, K., 2002b, The product line mapping approach to defining and structuring product portfolios, Proceedings IEEE Joint International Conference on Requirements Engineering, Los Alamitos, CA, USA, IEEE Comput. Soc
53	Birk, A., Heller, G., John, I., Schmid, K., Von der Massen, T. and Muller, K., 2003, Product Line Engineering: The State of the Practice, IEEE Software, 20, 6, 52-60
54	Aoyama, M., 2005, Persona-and-scenario based requirements engineering for software embedded in digital consumer products, Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on,
55	Mohan, K. and Ramesh, B., 2003, Ontology-based support for variability management in product and families, Proceedings of the 36th Annual Hawaii International Conference on System Sciences,
56	Buhne, S., Lauenroth, K. and Pohl, K., 2005, Modelling requirements variability across product lines, Proceedings of 13th IEEE International Conference on Requirements Engineering, Los Alamitos, CA, USA, IEEE Comput. Soc
57	Redling, T. J., 2003, A methodology for developing new product line requirements through gap analysis, Proceedings of 22nd Digital Avionics Systems Conference., Piscataway, NJ, USA, IEEE
58	Hammouda, I., Hautamaki, J., Pussinen, M. and Koskimies, K., 2005, Managing variability using heterogeneous feature variation patterns, FASE 2005 Held as Part of the Joint European Conference on Theory and Practice of Software, ETAPS 2005., Berlin, Germany, Springer-Verlag
59	Kim, M., Park, S., Sugumaran, V. and Yang, H., 2007, Managing requirements conflicts in software product lines: A goal and scenario based approach, Data and Knowledge Engineering, 61, 3, 417-432
60	Reiser, M. O. and Weber, M., 2006, Managing Highly Complex Product Families with Multi-Level Feature Trees, 14th IEEE International Conference on Requirements Engineering, ,
61	Rabiser, R. and Dhungana, D., 2007, Integrated support for product configuration and requirements engineering in product derivation, 33rd Euromicro Conference on Software Engineering and Advanced Applications, Piscataway, NJ, USA, IEEE

62	Savolainen, J., Kauppinen, M. and Mannisto, T., 2007, Identifying key requirements for a new product line, Proceedings of Asia-Pacific Software Engineering Conference, APSEC, Nagoya, Japan, IEEE Computer Society
63	Clements, P. C., Jones, L. G., McGregor, J. D. and Northrop, L. M., 2006, Getting there from here: A Roadmap for software product line adoption, Communications of the ACM, 49, 12, 1183261
64	Lee, J. and Muthig, D., 2006, Feature-oriented variability management in product line engineering, Communications of the ACM, 49, 12, 1183266
65	Mei, H., Zhang, W. and Gu, F., 2003, A Feature Oriented Approach to Modeling and Reusing Requirements of Software Product Lines, Proceedings of IEEE Computer Society's International Computer Software and Applications Conference, Institute of Electrical and Electronics Engineers Computer Society
66	Böckle, G., 2000, Model-based requirements engineering for product lines, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers
67	Hein, A., Schlick, M. and Vinga-Martins, R., 2000, Applying feature models in industrial settings, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers
68	Kishi, T. and Noda, N., 2000, Aspect-oriented analysis for product line architecture, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers
69	Fey, D., Fajta, R. and Boros, A., 2002, Feature Modeling: A Meta-Model to Enhance Usability and Usefulness, Proceedings of the Second International Conference on Software Product Lines, Springer-Verlag
70	Ferber, S., Haag, J. and Savolainen, J., 2002, Feature Interaction and Dependencies: Modeling Features for Reengineering a Legacy Product Line, Proceedings of the Second International Conference on Software Product Lines, Springer-Verlag
71	Mannion, M., 2002, Using First-Order Logic for Product Line Model Validation, Proceedings of the Second International Conference on Software Product Lines, Springer-Verlag
72	Kishi, T., Noda, N. and Katayama, T., 2002, A Method for Product Line Scoping Based on a Decision-Making Framework, Proceedings of the Second International Conference on Software Product Lines, Springer-Verlag
73	Simon, D. and Eisenbarth, T., 2002, Evolutionary Introduction of Software Product Lines, Proceedings of the Second International Conference on Software Product Lines, Springer-Verlag
74	Claudia, F. and Ralf, H., 2004, Product line potential analysis, Third International Conference of Software Product Lines, Boston, MA, USA,
75	Eriksson, M., Börstler, J. and Borg , K., 2005, The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations, Ninth International Conference on Software Product Lines, Rennes, France,
76	Asikainen, T., Mannisto, T. and Soininen, T., 2006, A Unified Conceptual Foundation for Feature Modelling, Proceedings of the 10th International on Software Product Line Conference, IEEE Computer Society

77	John, I., Knodel, J., Lehner, T. and Muthig, D., 2006, A Practical Guide to Product Line Scoping, Proceedings of the 10th International on Software Product Line Conference, IEEE Computer Society
78	Chastek, G. J., Donohoe, P. and McGregor, J. D., 2007, A Production System for Software Product Lines, Proceedings of the 11th International Software Product Line Conference, IEEE Computer Society
79	Lee, K., Kang, K. C., Koh, E., Chae, W., Kim, B. and Choi, B. W., 2000, Domain-oriented engineering of elevator control software: a product line practice, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers
80	Ardis, M., Dudak, P., Dor, L., Leu, W.-j., Nakatani, L., Olsen, B. and Pontrelli, P., 2000b, Domain engineered configuration control, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers
81	Thiel, S. and Peruzzi, F., 2000, Starting a product line approach for an envisioned market: research and experience in an industrial environment, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers
82	Dager, J. C., 2000, Cummins's experience in developing a software product line architecture for real-time embedded diesel engine controls, Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions, Denver, Colorado, United States, Kluwer Academic Publishers
83	Steger, M., Tischer, C., Boss, B., Müller, A., Pertler, O., Stolz, W. and Ferber, S., 2004, Introducing PLA at Bosch Gasoline Systems: Experiences and Practices,
84	Kircher, M., Schwanninger, C. and Groher, I., 2006, Transitioning to a Software Product Family Approach - Challenges and Best Practices, Proceedings of the 10th International on Software Product Line Conference, IEEE Computer Society
85	Jensen, P., 2007, Experiences with Product Line Development of Multi-Discipline Analysis Software at Overwatch Textron Systems, Proceedings of the 11th International Software Product Line Conference, IEEE Computer Society
86	Ishida, Y., 2007, Software Product Lines Approach in Enterprise System Development, Proceedings of the 11th International Software Product Line Conference, IEEE Computer Society
87	Sellier, D., Benguria, G. and Urchegui, G., 2007, Introducing Software Product Line Engineering for Metal Processing Lines in a Small to Medium Enterprise, Proceedings of the 11th International Software Product Line Conference, IEEE Computer Society
88	Jepsen, H. P., Dall, J. G. and Beuche, D., 2007, Minimally Invasive Migration to Software Product Lines, Proceedings of the 11th International Software Product Line Conference, IEEE Computer Society
89	Tischer, C., Muller, A., Ketterer, M. and Geyer, L. A., 2007, Why does it take that long? Establishing Product Lines in the Automotive Domain, 11th International Software Product Line Conference, SPLC 2007.,

[1] 9, 11, 16, 33, 38, 40, 45, 46, 48, 50, 52, 56, 60, 61, 62, 63, 64, 66, 72, 84, 2, 3, 13, 14, 15, 18, 24, 28, 31, 49, 53, 67, 68, 70, 74, 77, 79, 80, 81, 82, 85, 86, 88, 89, 78, 69, 83, 87

---

[2] 1, 3, 4-14, 16-22, 25, 26, 28-46, 49-55, 57-59, 61-64, 73, 75-79, 84, 86, 87
[3] 2, 15, 23, 24, 27, 47, 48, 56, 60, 65-72, 74, 80-83, 85, 88, 89
[4] 1, 2, 8, 11, 12, 13, 31, 34, 40, 41, 43, 46, 49, 51, 53, 56, 59, 69, 70, 72, 74, 76, 77, 78, 80, 84, 85, 86, 88, 3, 24, 30, 37, 52, 87, 75
[5] 1, 2, 4-10, 13, 16-20, 22, 23, 25-29, 31-36, 39-43, 45-51, 53, 55-64, 66, 67, 69-74, 76-78, 80-86, 89, 15, 75, 3, 11, 12, 14, 21, 24, 30, 44, 88, 37, 38, 52, 54, 79, 87
[6] 9, 11, 16, 33, 38, 40, 45, 46, 48, 50, 52, 56, 60, 61, 62, 63, 64, 66, 72, 84
[7] 2, 3, 13, 14, 15, 18, 24, 28, 31, 49, 53, 67, 68, 70, 74, 77, 79, 80, 81, 82, 85, 86, 88, 89
[8] 78
[9] 69, 83, 87
[10] 3, 12, 20, 34, 43, 44, 58, 83, 90, 91, 97
[11] 3, 4, 5, 11, 14, 15, 16, 19, 24, 26, 30, 34, 37, 43, 46, 47, 48, 49, 51, 52, 53, 54, 55, 58, 65, 72, 75, 76, 77
[12] 18, 22, 33, 40, 50, 63, 64, 66, 67, 70, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89
[13] 1, 2, 6, 7, 8, 9, 10, 12, 13, 17, 20, 21, 25, 27, 28, 29, 31, 32, 35, 36, 39, 41, 44, 45, 56, 57, 59, 60, 61, 62, 71, 73
[14] 42, 68, 69, 74, 78
[15] 23, 38
[16] 1, 4, 9, 10, 15, 16, 17, 18, 19, 20, 22, 23, 28, 29, 30, 31, 32, 34, 35, 36, 38, 39, 43, 46, 47, 49, 50, 54, 57, 58, 60, 61, 62, 67, 68, 70, 74, 76, 78, 79, 80, 83, 85, 87, 89
[17] 2, 3, 5, 6, 7, 8, 11, 12, 13, 14, 21, 24, 25, 26, 27, 37, 41, 44, 45, 51, 52, 56, 59, 71, 72, 75, 81, 82, 86, 88
[18] 3, 24, 32, 49, 81, 88
[19] 38, 52, 54, 59, 67, 79, 87
[20] 11, 15, 75
[21] 1, 2, 8, 11, 12, 13, 31, 34, 40, 41, 43, 46, 49, 51, 53, 56, 59, 69, 70, 72, 74, 76, 77, 78, 80, 84, 85, 86, 88
[22] 3, 24, 30
[23] 37, 52, 87
[24] 75
[25] 1, 2, 4-10, 13, 16-20, 22, 23, 25-29, 31-36, 39-43, 45-51, 53, 55-64, 66, 67, 69-74, 76-78, 80-86, 89
[26] 15, 75

Chapter 3 - Systematic Review of Solutions Proposed for Product Line Economics

Mahvish Khurum, Tony Gorschek, Kent Petersson
Blekinge Institute of Technology, Department of Systems and Software Engineering,
S-372 25, Ronneby, Sweden
mkm@bth.se, tgo@bth.se, kent.petersson@bth.se

Abstract

This paper presents a systematic review of all the solutions addressing different economic aspect of software product lines (SPL), e.g. SPL adoption cost-benefits, PL architecture decisions, SPL test strategies project cost overrun and so on, presented for the years 2000 to 2007. The goal of the review is to analyze the level of empirical application and/or validation of the proposed solutions with the purpose of mapping maturity as well as to what extent proposed solutions might be proven in terms of usability and usefulness. The important finding of this review is that although many economic solutions for SPL have been proposed over the years, the absence of qualitative and quantitative results from empirical application and/or validation makes it difficult to evaluate the potential of proposed solutions with respect to their usability and/or usefulness in relation to industry adoption. Suggestions have been presented to improve the existing situation.

1. Introduction

Software product families have received significant attention from the software engineering community since the 1990s [1, 2] . The concept of product lines aims towards having a set of systems that share a common, managed set of features, which satisfy the particular needs of a market segment, developed from a common set of core assets in a certain given way [1]. The product line approach is recognized as a successful approach for reuse in software development [3] with the major benefits of product lines adoption reported as reduced time to market [4, 5], reduced cost [6] and improved quality [4, 6, 7]. For these reasons many companies developing software intensive products have either adopted or are considering the adoption of a SPL approach [5, 8].

For making rational decisions about product line processes and products, it is important to answer the question “Does it pay?” [9]. In order to make sound software engineering decisions it is required to understand the business impact of those decisions and vice versa [9]. Without economic evaluation of adopting product line processes or products, a company might end up in spending more and getting fewer benefits. Therefore, evaluating the economic viability of product line adoption is critical.

Several economic measurement and evaluation models/methods aimed at SPL have been presented in academia and as industry experience reports. However, in order to gauge the usability and usefulness of the proposed models/methods, it is important to see the empirical evidence of their application and/or validation, e.g. in industry or through a simulation [10].

This paper presents a systematic review (based on the systematic review technique proposed by Kitchenham in [11]) conducted on the papers, which either proposed or reported on experience with economic/cost/cost-benefit/ROI models/methods (called *solutions* from now on) for SPL. The motivation was to gauge the level of actual industry adoption, i.e. to what extent the presented solutions are applied and/or validated in industry. In addition to industry validation all other types of empirical results are collected and taken into consideration to offer a detailed summary of the empirical evidence available. To achieve this, the selected papers are categorized and analyzed from several perspectives, such as research basis, method, level of validation and type of empirical results in relation to usability and usefulness of the proposed solutions. For industry practitioners, looking to perform economic evaluations using a certain solution, the results of the study can be used as an indication of maturity as well as to estimate potential risk of a certain solution. From an academic point of view researchers planning studies and evaluation of a solution can use this study as inspiration for study design as the evaluation criteria of the review presented in this paper could be seen as a checklist to test a solution's usability and usefulness.

2. Related Work

Four previous studies ([12]; [13]; [14] and [15]) have reviewed the state of empirical research in different areas e.g. computer science, software engineering, web engineering and so on. However, to the best of our knowledge no other study has been conducted with the same focus as the review presented in this paper. This review does not aim to systematically classify proposed economic solutions as methods, models, tools, framework or classify the papers according to the classification and evaluation scheme suggested by W. Roel et al. [16]. The goal of this review is to analyze practical application and validation of proposed solutions in industry to gauge their practical usability and usefulness. In addition to this other empirical evidence is also considered e.g. proof of usability and/or usefulness demonstrated through simulation or experimentation.

3. Design

3.1. Research Questions and Definitions

The four research questions driving the systematic review can be viewed in Table 1.

.

Table 1 – Research questions and motivation.

Research Questions and Motivation
RQ1. Are proposed solutions based on needs identified in industry? **Motivation:** Is the solution presented based on any need/issue/problem (called a need from here onwards) identified in industry though empirical investigation? Examples can be process assessments, case studies, participation knowledge, surveys, observations, and so on. Both direct and indirect sources will be considered, giving the presented studies the benefit of the doubt.
RQ2. Are proposed solutions applied and/or validated in laboratory setting or industry? **Motivation:** Is the solution presented applied/validated through simulation or in industry as a part of the presentation in the paper? Any validation in industry from static validation to dynamic validation will be considered [17].
RQ3. Are the proposed solutions usable? **Motivation:** If the authors of a proposed have applied/validated it in industry or demonstrated its application/validation through simulation, identify any indications/proof/reports on its usability in the venue of the application/validation (usability is defined below).
RQ4. Are the proposed solutions useful? **Motivation:** If the authors of a proposed have applied/validated it in industry or demonstrated its application/validation through a controlled experiment, identify any indications/proof/reports on its usefulness in the venue of the application/validation (usefulness is defined below).

The two main terms used in the research questions, namely *usability* and *usefulness*, are defined and exemplified below.

Usability is defined in terms of;

- Scalability of Introduction. How scalable the proposed solution is in terms of its introduction and application cost including e.g. which metrics have to be collected, who collects the metrics?

- Scalability of Use. How scalable is the proposed solution in terms of metrics collection and recording? For example, if a cost-benefit estimation model is proposed and certain data is required for making estimations, do the defined metrics provide the data required even when the size of a product grows and multiple products come in to play, or does it make simplified estimations with simple data collected or is there any indication that product line grade scalability is possible (or even considered by the creators of the solution)?

Scalability of Introduction and Scalability of Use point to a micro quality of a solution and that is its efficiency. If a proposed solution is demonstrated to have any of these aspects of efficiency, the corresponding paper is counted as having some proof of usability.

Usefulness is defined in terms of;

- Better Alternative Investment. For example, a proposed solution (X) is better than an alternative (maybe previously used) solution (Y), and/or,

- Effectiveness. The effectiveness of a proposed solution in relation to achieving goals or solving the problems it was designed for. For example, solution X makes estimations 15% more realistic than before.

Again, a solution demonstrating either of the two aspects of usefulness is counted as having some proof of usefulness.

3.2. Review Design

In this section, the systematic review design is presented describing paper identification method, inclusion/exclusion criteria, and the classification scheme.

3.2.1. Search Strategy. The search string used to identify literature for the mapping study is given below:

((("software product line" AND economic) WN KY) AND (2000-2007 WN YR)) or ((("software product line" AND (ROI OR {return on investment})) WN KY) AND (2000-2007 WN YR)) or (((software NEAR/0 product NEAR/0 lin* AND cost NEAR/0 model*) WN KY) AND (2000-2007 WN YR)) or ((("software product line" AND economic* NEAR/0 model*) WN KY) AND (2000-2007 WN YR)) or ((("software product line" AND {value based software engineering}) WN KY) AND (2000-2007 WN YR))*

When applying the search string, the aim was to identify highly relevant sample of product line literature by focusing on the Software Product Line Conferences (SPLC) and the Workshops on Product Family Engineering (PFE). SPLC is the premier forum for practitioners, researchers, and educators presenting and discussing experiences, ideas, innovations, as well as challenges in the area of SPL. SPLC also has a relatively large industry presence. In order to identify the articles related to SPL economics' aspects, the search string was applied on all papers in all the previous SPLC and PFE proceedings (2000-2007). To further extend the study papers from international, peer-reviewed journals have also been included. The databases that searched are Inspec, Compendex, ACM Digital Library and IEEE. In total, 79 papers were identified through the searches.

3.2.2. Identification and Selection of Papers. The procedure used for the identification of relevant papers (out of the 79 generated from the search string) can be described as follows. The Title, Abstract and Conclusions were read for all the 79 papers. Only if none of the parts contained any potentially relevant information within the scope of the review was a paper dismissed. This means that if there was any new proposed economic solution for SPL or any experience report indicating use of any SPL economic solution, it was included otherwise discarded.

The inclusion and exclusion criteria were pilot-tested by the authors on a random sample of 10 papers. An agreement on inclusion and exclusion was achieved. After the pilot, the primary author screened the remaining papers and marked them as included/excluded based on the approach described.

3.2.3. Data Extraction Procedure. Based on the research questions identified (see Section 3.1.), a set of data extraction categories was identified. The categories were identified through using the Goal Question Metric (GQM) approach [18] during several brain storming sessions to ensure that categories identified addressed the aspects required to answer the research questions. Due to reasons of brevity, the data extraction categories are not included but can be made available upon request to the authors.

The categorization of quality attributes (usability and usefulness) into quantitative and qualitative is not intended to indicate a preference or valuation of one over the other. Any empirical data is judged on its own merits. For example, quantitative results obtained through a simulation might not be as valuable as the expert opinion obtained in a case study with industry practitioners who actually applied a particular solution in industry. Moreover, context, background description and design also weigh in as the purpose is to categorize the reported empirical data to analyze the levels of usability and usefulness of a proposed solution. For example, a claim about the usability and/or usefulness of the presented solution without any description of context or how the claim can be substantiated is still considered as empirical evidence from the perspective of the study, but further analysis lets the reader weigh the value of the evidence.

Similar to the inclusion/exclusion process, the data extraction process was tested by the authors using a sample of 7 included papers. This was done to ensure that there was a common understanding of the categories defined and the classification was agreed upon by two researchers avoiding the potential bias and error source of having only one researcher performing the categorization.

4. Results and Analysis

Out of 79 papers found through the search string (see Section 3.2.1), 19 papers [9, 19-36] were selected. The selected papers are analyzed with respect to the research questions posed in Table 1.

4.1. RQ1 (Are proposed solutions based on needs identified from Industry?)

All the selected papers are based on the need identified from industry except [20, 21, 23, 27, 35]. However, a deeper analysis of the empirical basis reported can be seen in Table 2 which shows that a majority of the papers have mentioned identified needs as "Statements only" ([9, 19, 20, 24-26, 28, 29, 31, 33, 36], 78.57%), or as "Participation knowledge" ([21, 22, 32, 34], 28.57%). None of the papers have mentioned e.g. interviewing experts to identify needs, or using some form of process assessment in industry to identify the need for the proposed product line economics' solutions. The technical reports referenced in some papers were also read but empirical process used to identify the need was not found. These results make it hard to judge the credibility of the empirical basis of the solutions proposed due to the absence of presentation of e.g. process assessment and/or any other empirical study. In addition, due to the almost total lack of how the practitioners knew about the needs that constitute the basis for the solutions proposed, it is impossible to draw any conclusions.

Moreover, it is found that although a majority of the papers claims empirical basis, only 4 [28, 31, 32, 35] papers out of 19 are based on future work described by previously published papers, or extend previously published solutions. This may indicate that in the absence of a proper empirical investigation investigating the needs in industry, the needs quoted may not be representative or valid for other companies in similar situations.

The answer to RQ1 is that a majority of the proposed solutions are based on needs identified in industry, however, the actual method used and the validity of the results is impossible to ascertain as very little information is given.

4.2. RQ2 (Are proposed solutions applied and/or validated in laboratory setting or industry?)

In order to answer this question, let us do an analysis of the application/validation methods used and the explanation of results obtained. An application/validation could range from controlled experiments/simulations to full-fledged industry pilots.

In total, 14 papers [21-32, 34, 35] out of 19 have performed some application/validation of the proposed solutions. This seems to be a positive thing but a deeper analysis of the papers, claiming application/validation, reveals that only 6 ([21, 22, 28, 29, 32, 34], 42.85%) out of these 14 have used case study as an application/validation method whereas the remaining papers (57.14%) have demonstrated application/validation through simplified examples using simplified or fictitious data. This makes it harder to judge the scalability of introduction and scalability of use of the proposed solutions.

Empirical studies are the building blocks essential for collecting evidence and to determine what situations are best for using a particular solution [37]. The current situation means there is a lack of quantitative or qualitative data that a new solution is better than an already existing one, or what the impact of implementing the new solution might be. This makes it impossible to gauge efficiency or effectiveness of proposed solutions either alone or in relation to better alternative investment (BAI).

Table 2 - Selected papers, "Basis" categorization.

Basis Reported as	Total
Statements only	11
Participation knowledge	4
Interviews	0
Process assessment	0
Total	15

Moving on from the analysis of application/validation methods used to the analysis of the application/validation design details, Table 3 shows the categorization of the application/validation design explanation given in the selected papers. It can be seen that some form of application/validation was performed. By examining the last column it is possible to see that 11 out of 14 papers (78.57%) either provide a short summary ([21, 22, 24, 25, 34, 35]) or explain application/validation in detail ([28-32]). This seems to be a positive outcome, i.e. most of the papers have explained the application/validation in detail. However, after analyzing the level of application/validation results (see Table 4), it can be seen that a majority of the papers ([21, 23, 25, 26, 30-32, 34, 35]) have only statements about the application/validation results. Only 2 papers ([28, 29]) provide qualitative results and only 2 papers ([22, 24]) provide quantitative results.

Table 3 - "Application/Validation Design Explained" categorized.

Application/Validation Design Explained	Total
Statements only	3
Application/Validation summary	6
Application/Validation in detail	5
Total	14

Figure 1 presents another aspect in relation to RQ2, i.e. how many solutions from each year are based on solutions presented previous years. Figure 1 shows that many new solutions have been presented over the years, but very few actually have been used as a basis for further development or adopted in industry. For example, by the year 2005 a total of 10 new product line economics' solutions had been proposed but none of the papers reported use of any of the previously proposed solutions (in industry or as a basis for refinement of a solution). By 2007 the number of new solution papers had reached 19, but only 4 papers ([28, 31, 32, and 35]) were based on previously proposed solutions. The technical reports referenced in some papers were also read but qualitative and quantitative results were not found.

Table 4 - "Application /Validation Results" categorized.

Application/Validation Results Explained	Total
Nothing	1
Statements only	9
Results as experts opinion	2
Quantitative data	2
Qual +Quant	0
Total	14

This can indicate that the proposed solutions are not applicable in industry or that due to missing application/validation results the solutions are not applied by practitioners and not used by researchers. This problem has been indicated by others as well e.g. in [38]. This can imply that a focus on validation and proper reporting should be premiered over the continuous presentation of new solutions. Another possibility is that the proposed solutions do not solve the challenges in industry, which in turn implies that there is a need to understand the challenges. Another possible conclusion could be that industry practitioners are not up to date with the new solutions proposed, thus the solutions go unused. None of the papers reported replicated studies.

Summarizing all the aspects analyzed above, the answer to RQ2 is that although 73.68% of the proposed solutions from 2000 to 2007 have reported some form of application/validation, the absence of detailed results other than statements and the lack of replicated studies make it difficult to evaluate the potential of the proposed solutions.

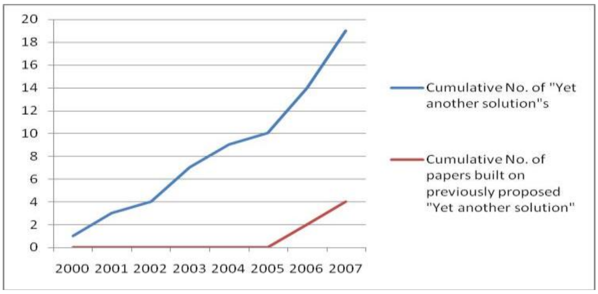

Figure 1 - Cumm No. of "New Solution" VS Cumm No. of papers built on previously proposed solutions.

4.3. Answering RQ3 (Are the proposed solutions usable?) and RQ4 (Are the proposed solutions useful?)

By looking at Table 5 it is possible to see that 15 out of 19 selected papers have not stated anything about usability in terms of scalability of introduction or scalability of use. 4 papers out of 19 ([24, 27, 28, 30], 21.05%) have only statements claiming usability. For example, *"It was hard to collect, for example, the cost of a typical product released using the traditional style. In such cases, apply well-known prediction models and validate with managers.*"[28]. None of the papers have given qualitative and quantitative evidence of usability.

Clearly with the usability statements as exemplified above, it is difficult to judge the usability of a proposed solution. A majority of the papers do not include either qualitative or quantitative data in relation to solution scalability of introduction or scalability of use, making it harder for practitioners to evaluate usability. It is important to understand that the intention in this review is not to criticize papers but to highlight the absence of qualitative and quantitative evidence in relation to usability, which might be a barrier for the industrial adoption of the proposed solutions.

Table 5 - "Usability" reported

Usability	Total
Statements only	4
Results as experts opinion	0
Quantitative results	0
Qual +Quant	0
Total	4

Looking at the "Total" column in Table 6, 2 ([20, 34], 10.52%) out of 19 papers have not mentioned anything about usefulness. The remaining 17 papers have reported positive results which seem to be very good, but a deeper analysis shows that these 17 papers out of 19 have claims of usefulness as statements only. For example "*We feel that the uncertainty analysis can reveal more insights about the ROI in a product line, and can give statistical evidence to the managers about investing into a product line*" [28] or "*we regard the economic model we are about to develop as a key step towards sound engineering for product line* *development*" [23]. There is not a single paper that provides qualitative and/or quantitative data about usefulness. The technical reports referenced in some papers were also read but qualitative and quantitative evidence of usability and/or usefulness were not obvious.

Table 6 - "Usefulness" reported

Usefulness	Total
Statements only	17
Results as experts opinion	0
Quantitative results	0
Qual +Quant	0
Total	17

The answer to RQ3 and RQ4 is that although there are statements about usability and usefulness in the papers published between the years 2000 to 2007, a lack of qualitative and quantitative evidence makes it difficult to evaluate how usable and useful the proposed solutions and collected experiences are. In industry, time and resources are scarce. If a practitioner cannot in any manner determine the time and resources required to implement a solution, and at least approximate the possible usefulness of the solution in comparison to available better alternative investments, it is very unlikely that the solution will be adopted. Similarly, if authors do not show any evidence regarding scalability of use of a particular solution for industry grade problems, practitioners would probably not take the risk of implementing a solution falling short on reporting even rudimentary evidence of efficiency and effectiveness.

5. Suggestions

There are several lessons from Software Engineering research and practice which Software Product Line community can learn from.

5.1. Conduct Process Assessment

Instead of stating that that solution is based on participation knowledge a light-weight software process assessment methods (e.g. [39]) can be employed to ensure that the need identified is triangulated and has empirical basis. This can help research community to investigate the real needs which can consequently result in further research papers based on the needs identified. Industry would benefit from the proposed solution for the need identified as it would be applicable for the real problems faced in industry.

5.2. Conduct Empirical Studies

Once a solution is proposed (based on the empirical need identified through process assessment) small controlled and well planned experiments/simulations and replications with real data, even if conducted offline can be valuable. The potential benefits of such studies include: getting initial evidence to confirm or refute hypotheses, controlling factors that can affect the study, proving the relevance of the research to the research community and industry, fine-tuning the proposed method before full-fledged industry pilot. After the initial evaluation showing positive results the validation can be taken one step further by conducting a case study in industry. This could be used to fine-tune the proposed solution before a pilot could actually be implemented. Inspirations can be taken from example papers like [14, 28, 29] and [22, 24] as how to design studies and collect and present the real qualitative and quantitative empirical results. A model for technology transfer [17] can be used to design the validation steps. Due to complexity involved in designing empirical studies for SPL area in general and SPL economic solutions in particular, it is necessary to identify the limitations and customize the existing empirical methods for proposed economic solutions rather than having only statements about empirical evidence.

5.3. Identify Metrics

It is understandable that in case of SPL, the complexity of conducting empirical studies increases but to deal with this, an important aspect is that proper metrics are identified to measure different aspects of proposed economic solutions. The definitions of usability and usefulness stated in this paper can be used as a starting point to identify what metrics to collect and report when claiming usability and usefulness of the proposed SPL economic solutions. Some of the example papers that employed such metrics in their empirical studies are [40], [41]. Collecting data for the identified metrics during the empirical evaluation would greatly enhance the quality of reported usability and usefulness evidence both for research community for further research and for industry considering adoption of a particular solution.

6. Conclusion

This paper presents the systematic review of all the papers, proposing solutions for SPL economical aspects, from the year 2000 until 2007. After screening 79 papers published, 19 papers were selected, that discuss different aspects of product line economics, to *analyze the practical application and validation of proposed solutions in industry to gauge their practical usability and usefulness,* four research questions were specified (see Section 3.1.). Based on the goal and corresponding research questions to achieve that goal, a data extraction procedure was defined. Against the defined procedure, data was extracted from the included papers about the basis, practicality, usability and usefulness, future work and authors.

The major findings of the review can be summarized as follows:

1. ***Issue:*** Many solutions discussing economics aspects of product lines have been presented over the years and majority of the papers address needs identified in industry, but they fall short on the strength of the approach used to identify the need for a solution. Most papers only claim that they based the solution on a need identified in industry or state that through participation knowledge they found the need for the proposed solution. ***Consequences:*** Such claims and statements may be valid, but they raise validity questions both from a research perspective and an industrial adoption perspective. Without interviewing experts in industry or performing some form of process assessment, it is hard to triangulate the need identified thus raising the issue that the need may not be representative of the current situation. As a result, this poses questions about the internal and external validity of the needs identified and corresponding solutions proposed. ***Suggestions:*** conduct process assessment (see Section 5.1.)

2. ***Issue:*** Many papers claim that they have applied/validated the proposed solutions in some form; however, a deeper analysis reveals that a majority of the claims are merely statements (see Table 4) and qualitative and quantitative evidence supporting these claims is generally missing. ***Consequence:*** Claims and statements may be valid, but in the absence of clear qualitative evidence as experts' opinions and/or quantitative data (by using real data in simulations) about the benefits of the proposed solution, it is hard to evaluate the potential of these solutions for industry adoption. ***Suggestions:*** Conduct empirical studies (see Section 5.2.).

3. ***Issue:*** Papers claim usability and usefulness of the proposed solutions in some form, however a deeper analysis reveals that majority of the claims are also merely statements about usability and usefulness (see Table 5 and Table 6). ***Consequence:*** As mentioned previously such claims may be valid, but they raise validity questions from both a research and industrial adoption perspective. Without experts' opinions and/or quantitative data supporting the usability and usefulness claims, it is difficult to evaluate the validity of the claims, and similarly it is difficult for the practitioners to evaluate the usability and usefulness of a proposed solution for application in industry. ***Suggestions:*** identify metrics (see Section 5.3.).

The overall goal of this review was not to expect or demand perfect evidence of usability and usefulness following perfect and extensive data collection in industry. However, many studies over the years have shown that it is possible to validate solution suggestions in any number of ways. Simulation could be used in academia with real data. Static (preliminary) validation can be performed in industry through workshop, interviews, or surveys. Dynamic validation (e.g. pilots) can be performed collecting metrics and qualitative data though interviews with practitioners [42]. The data collected is not complete, but vastly better than no data at all.

In addition to doing validation (e.g. in industry), the way in which the validation is planned and reported is also crucial. The papers reviewed are full of statements, claiming usability and usefulness. The good thing is that this indicates that our interest in these two concepts in this systematic review is relevant, i.e. usability and usefulness of solutions are important. However, even if statements are common, no evidence is presented. Both in terms of absence of data, but also absence of design for the studies presented. The only seemingly complete validation is when there is no real validation, e.g. in case of presenting simplified examples. The use of simplified examples is not without merit, e.g. it can be used to explain and exemplify the use of a method, but the use of a simplified example is not the same as validation, but at least it could be made better with real data to demonstrate the usefulness of the solution.

This review does not attempt to say that the proposed solutions have no empirical application/validation results and empirical usability and usefulness data instead it highlights that this evidence needs to be made explicit to encourage industry adoption. The presence of empirical evidence of any sort with at least some intent to explain the overall design and execution of a validation (e.g. a pilot test in industry) could be very beneficial for both researchers and industry practitioners. From an academic point of view the possibility to learn and extend on presented research is crucial for progress. In addition, one of the foundations of research is the possibility to replicate studies. None of the papers included was a replicated study.

From a practitioner point of view, a design and illustration of how conclusions about usability and usefulness are made can vastly improve the relevance of any paper. The absence of data or evidence is problematic from two perspectives. First, can the results be trusted? Second, even if the authors are given the benefit of the doubt, is the proposed solution relevant for all cases?

6. References

[1] P. Clements and L. Northrop, Software product lines : practices and patterns. Boston MA: Addison-Wesley, 2002. 0-201-70332-7.

[2] S. Deelstra, S. Marco, and B. Jan, "Experiences in Software Product Families: Problems and Issues During Product Derivation," in *Proceedings of Third International Conference, SPLC2004* Boston, MA, USA: Springer, 2004.

[3] M. Kim, S. Park, V. Sugumaran, and H. Yang, "Managing requirements conflicts in software product lines: A goal and scenario based approach," *Data Knowl. Eng.,* vol. 61, pp. 417-432, 2007.

[4] W. A. Hetrick, C. W. Krueger, and J. G. Moore, "Incremental return on incremental investment: Engenio's transition to software product line practice," in *Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications* Portland, Oregon, USA: ACM, 2006.

[5] J. C. Dager, "Cummins's experience in developing a software product line architecture for real-time embedded diesel engine controls," in *Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions* Denver, Colorado, United States: Kluwer Academic Publishers, 2000.

[6] K. Pohl, G. Boeckle, and F. van der Linden, *Software Product Line Engineering*: Springer, 2005. 10 3-540-24372-0 Springer Berlin Heidelberg New York.

[7] M. Staples and D. Hill, "Experiences adopting software product line development without a product line architecture," in *Software Engineering Conference, 2004. 11th Asia-Pacific,* 2004, pp. 176-183.

[8] G. Bockle, "Model-based requirements engineering for product lines," in *Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions* Denver, Colorado, United States: Kluwer Academic Publishers, 2000.

[9] R. S. Faulk, R. Harmon R., and D. Raffo M., "Value-Based Software Engineering," in *Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions* Denver, Colorado, United States: Kluwer Academic Publishers, 2000.

[10] D. I. K. Sjoeberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N. K. Liborg, and A. C. Rekdal, "A survey of controlled experiments in software engineering," *Software Engineering, IEEE Transactions on,* vol. 31, pp. 733-753, 2005.

[11] B. A. Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering," in *Technical Report EBSE-2007-01*, 2007.

[12] V. Ramesh, L. Glass Robert, and I. Vessey, "Research in computer science: an empirical study," *The Journal of Systems and Software,* vol. 70, pp. 165-176, 2004.

[13] R. L. Glass, I. Vessey, and V. Ramesh, "Research in software engineering: An analysis of the literature," *Information and Software Technology,* vol. 44, pp. 491-506, 2002.

[14] E. P. Dewayne, A. P. Adam, and G. V. Lawrence, "Empirical studies of software engineering: a roadmap," in *Proceedings of the Conference on The Future of Software Engineering* Limerick, Ireland: ACM, 2000.

[15] E. Mendes, "A systematic review of Web engineering research," in *Empirical Software Engineering, 2005. 2005 International Symposium on*, 2005, p. 10 pp.

[16] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion," *Requir. Eng.,* vol. 11, pp. 102-107, 2005.

[17] T. Gorschek, C. Wohlin, P. Garre, and S. Larsson, "A model for technology transfer in practice," *IEEE Software,* vol. 23, pp. 88-95, Nov-Dec 2006.

[18] V. Basili, G. Caldiera, and H. D. Rombach, *The Goal Question Metric Approach*: John Wiley & Sons, 1994.
[19] S. Klaus, "An Initial Model of Product Line Economics," in *Revised Papers from the 4th International Workshop on Software Product-Family Engineering*: Springer-Verlag, 2002.
[20] P. Knauber, J. Bermejo, G. Bockle, J. C. S. d. P. Leite, F. v. d. Linden, L. M. Northrop, M. Stark, and D. M. Weiss, "Quantifying Product Line Benefits," in *Revised Papers from the 4th International Workshop on Software Product-Family Engineering*: Springer-Verlag, 2002.
[21] K. Schmid and M. Verlage, "The Economic Impact of Product Line Adoption and Evolution," *IEEE Softw.*, vol. 19, pp. 50-57, 2002.
[22] K. Schmid, "A Quantitative Model of the Value of Architecture in Product Line Adoption," in *SPLC*, 2003.
[23] G. Böckle, P. Clements, J. D. McGregor, D. Muthig, and K. Schmid, "A Cost Model for Software Product Lines," in *Software Product - Family Engineering. 5th International Workshop* Siena, Italy: Springer Berlin / Heidelberg, 2003, pp. 310-316.
[24] D. R. Peterson, "Economics of Software Product Lines," in *Software Product - Family Engineering. 5th International Workshop* Siena, Italy: Springer Berlin / Heidelberg, 2003, pp. 381-402.
[25] B. Boehm, A. W. Brown, R. Madachy, and Y. Yang, "A Software Product Line Life Cycle Cost Estimation Model," in *Proceedings of the 2004 International Symposium on Empirical Software Engineering*: IEEE Computer Society, 2004.
[26] G. Bockle, P. Clements, J. D. McGregor, D. Muthig, and K. Schmid, "Calculating ROI for software product lines," *Software, IEEE*, vol. 21, pp. 23-31, 2004.
[27] S. B. A. B. Lamine, L. L. Jilani, and H. H. B. Ghezala, *Cost Estimation for Product Line Engineering Using COTS Components*, 2005.
[28] D. Ganesan, D. Muthig, and K. Yoshimura, "Predicting Return-on-Investment for Product Line Generations," in *Proceedings of the 10th International on Software Product Line Conference*: IEEE Computer Society, 2006.
[29] K. Yoshimura, D. Ganesan, and D. Muthig, "Defining a strategy to introduce a software product line using existing embedded systems," in *Proceedings of the 6th ACM; IEEE International conference on Embedded software* Seoul, Korea: ACM, 2006.
[30] M. Nonaka, L. Zhu, and M. Ali Babar, *Project Cost Overrun Simulation in Software Product Line Development*, 2007.
[31] M. Nonaka and L. Zhu, "Impact of Architecture and Quality Investment in Software Product Line Development," in *Proceedings of the 11th International Software Product Line Conference*: IEEE Computer Society, 2007.
[32] D. Ganesan, J. Knodel, R. Kolb, U. Haury, and G. Meier, "Comparing Costs and Benefits of Different Test Strategies for a Software Product Line: A Study from Testo AG," in *Proceedings of the 11th International Software Product Line Conference*: IEEE Computer Society, 2007.
[33] P. C. Clements, "An Economic Model for Software Architecture Decisions," in *Proceedings of the First International Workshop on The Economics of Software and Computation*: IEEE Computer Society, 2007.
[34] Y. Matsumoto, "A Guide for Management and Financial Controls of Product Lines," in *Proceedings of the 11th International Software Product Line Conference*: IEEE Computer Society, 2007.
[35] H. P. In, J. Baik, S. Kim, Y. Yang, and B. Boehm, "A quality-based cost estimation model for the product line life cycle," *Commun. ACM*, vol. 49, pp. 85-88, 2006.
[36] H. Schackmann and H. Lichter, "A Cost-Based Approach to Software Product Line Management," in *International Workshop on Software Product Management, 2006. IWSPM '06.*, 2006, pp. 13-18.
[37] S. L. Pfleeger, "Albert Einstein and empirical software engineering," *Computer*, vol. 32, pp. 32-38, 1999.
[38] M. Kircher, C. Schwanninger, and I. Groher, "Transitioning to a Software Product Family Approach - Challenges and Best Practices," in *Proceedings of the 10th International on Software Product Line Conference*: IEEE Computer Society, 2006.
[39] T. Gorschek, "Software Process Assessment & Improvement in Industrial Requirements Engineering," in *School of Engineering - Department of Systems and Software Engineering* Ronneby: Blekinge Institute of Technology, 2004, p. 154.

[40] E. Magnus, B. Jürgen, and B. Kjell, "The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations," in *Ninth International Conference on Software Product Lines* Rennes, France, 2005.
[41] T. Gorschek, M. Svahnberg, A. Borg, A. Loconsole, J. Borstler, K. Sandahl, and M. Eriksson, "A controlled empirical evaluation of a requirements abstraction model," *Information and Software Technology*, vol. 49, pp. 790-805, 2007.
[42] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "Industry evaluation of the Requirements Abstraction Model," *in print Requirements Engineering*.

Chapter 4 - A Method for Early Requirements Triage and Selection Utilizing Product Strategies

Mahvish Khurum, Khurum Aslam and Tony Gorschek
Blekinge Institute of Technology, Department of Systems and Software Engineering,
S-372 25, Ronneby, Sweden
mkm@bth.se, aslam@spidexa.com, tgo@bth.se.

Abstract

In market-driven product development large numbers of requirements threaten to overload the development organization. It is critical for product management to select the requirements aligned with the overall business goals and discard others as early as possible. This paper presents a method that utilizes strategies for early requirements triage. The method presented is based on good practices from management literature customized for technical product strategy formulation. The creation of the method was also inspired by industry needs through exploratory interviews covering requirements triage and selection processes as well as strategy formulation. After creation, the method was subsequently validated in industry where the aspects of usability and usefulness were investigated, yielding promising initial results.

1. Introduction

Due to the emergence of markets for off-the-shelf or packaged software [1, 2], market-driven development is gaining increased interest in comparison to customer-specific system development [3, 4]. As a consequence a shift in focus is occurring, affecting software development in general and requirements engineering in particular [4]. In contrast to traditional requirements engineering, requirements in market driven requirements engineering (MDRE) come from internal sources like developers, marketing groups, sales teams, support groups, bug reports, as well as from external sources such as different users, customers groups from different and multiple market segments, and competitors (through e.g. surveys, interviews, focus groups, and competitor analysis) [5]. The result is a large amount and continuous flow of requirements that threaten to overload the development organization [3].

In addition, MDRE activities are not limited to one development instance but play their role continuously as a part of product management as products are developed and new releases come over time [3, 31]. Research and industry reports put emphasis on the selection of requirements based on product strategies, business goals and the overall vision of an organization as it enables optimizing both long term and short term perspectives as well as aligning the whole organization towards the same direction [4, 8, 16].

From an MDRE perspective it is important to be able to handle large amounts of requirements continuously, performing early triage of requirements, selecting the ones aligned with a specific product's strategy. Early triage can be described as discarding "inappropriate" requirements at the product management level as early as possible, without expending large resources for the triage activity as the process has to handle large amounts of requirements.

However, in industry, while managers regard strategy formulation and use as the most important aspect of technology management [17], strategy formulation is usually performed ad-hoc, and a systematic approach for formulating strategies is often missing in practice [18]. Even if the formulation of strategies was pursued, the factors affecting strategy formulation differ between different stakeholders. Strategic and middle management and technical experts all need to share *one* vision. Strategic managers often overlook the technical perspective, and technical experts can be unaware of or overlook the strategic managers' perspective. This paper explores how the strategic and technical perspectives can be combined to formulate product strategies that are good-enough for early requirements triage and selection. A Method for Early Requirements Triage and Selection (MERTS) is presented. The method has two main purposes. First, it acts as a stepwise guide to creating product strategies taking both strategic and technical views into account. To the best of our knowledge MERTS is the first method of its kind in the area of software engineering. Vähäniitty [21] suggested "key decision areas" to be incorporated in a strategy but description of what factors to be included for each of the decision areas explicitly is rather abstract and how to weigh their relative importance is missing. Secondly, the strategies resulting from MERTS can be used by managers to perform requirements triage, in essence selecting the "right" requirements for realization. The creation of MERTS was preceded by the study of traditional management literature covering strategy formulation, and complemented with exploratory interviews in industry utilizing two technology driven companies working in a market driven context. The goal was to develop a method that was usable and useful in a real industry context taking the companies limited time and resources to perform strategy formulation, triage and selection into account. Following the creation of MERTS the method was tested by interviewing industry practitioners to get an initial validation of the method's usability and usefulness in industry.

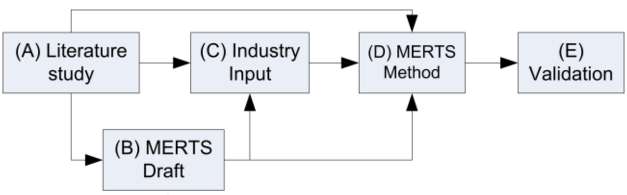

Fig 1 – Activities involved in MERTS creation

Fig 1 shows the activities involved in the creation of MERTS. Section 2 of this paper (A) gives an overview of requirements triage and product strategies formulation based on requirements engineering and product management literature study. Based on the literature study an initial draft of MERTS was created (B). Following this we elicited information from industry. The purpose of industry input (C) was twofold: (I) explore the requirements triage and product strategies creation process in industry and (II) ensure that the basic MERTS draft targeted the real issues in industry regarding the use of product strategies and requirements triage. Therefore, an semi-structured interviews were performed based on the literature study and the initial MERTS draft to get input from industry. For detailed process refer to [23].

Section 3 (C) describes the initial input from industry. The input from the previous literature study (A) and industry interviews (C) was subsequently used for the creation of a final version of MERTS (D) which is presented in Section 4 along with the prerequisite and steps inherent to the method. In Section 5 (E), initial industry validation is presented and a discussion of lessons learned highlighting initial feedback is presented. Section 6 concludes the paper and discusses directions for future work.

2. Related Areas and Literature Study

2.1. Requirements Triage

”Triage” is a term used by medical practitioners to describe and prioritize treatment of patients based on their symptoms severity [14]. Triage attempts to optimize resource usage by balancing the need for treatment with the likelihood of a successful outcome [14]. In order to do this successfully a patient is assessed on criteria like “physiology”, “injury anatomy”, “mechanism of injury” and a decision is made whether to treat the patient, or in case of multiple patients, in what order to treat the patients. Inspired by this approach, Simmon [14] has suggested to measure different aspects of requirements e.g. requirements implementation cost, volatility, schedule slips, lack of domain experience to evaluate a requirement and decide about its selection or rejection.

Davis [10] suggests prioritizing requirements based on their interdependencies and estimated effort required, and then performing triage to select a set of requirements that optimizes the probability of product success in its targeted market.

The concept of triage is appealing as it addresses one of the main problems with MDRE, and can alleviate the risk of requirements overload.

2.2. Product Strategies

A broader definition of business strategy presented by Oliver 2000 [13] is as follows: “*the understanding of an industry structure and dynamics, determining the organization’s relative position in that industry and taking action either to change the industry’s structure or the organization’s position to improve organizational results*”. This is somewhat related to product strategy which is defined as: “*Product strategy begins with a strategic vision that states where a company wants to go, how it will get there, and why it will be successful*” [6].

In order to formulate a product strategy there are a number of questions that need to be answered. Below each key question is described and exemplified.

The question (1) “**where do we want to go**” requires finding out the right balance between the long term opportunities [6] or goals [7] and short term objectives [6]. The basic aim of the goals is to set the general directions of movement, whereas objectives specify specific measures of accomplishment [7]. Examples of goals are *profit*, *growth*, and *market share*. The product strategy normally has one of the above mentioned types of goals because these goals are potentially conflicting. If the goals/objectives revolve around increase in growth along with increase in profits, this it is unrealistic because to attain reasonable growth increase in expenditures or decrease in profit margins is required [12]. Therefore depending on the product life cycle stage, one of the goals would have a priority.

The answer to (2) "**how will we get there**" formulates the core of the product strategy [15]. It addresses aspects such as *customer targets, competitive targets,* and *differential advantage*.

The choice of *customer targets* depends on the nature of the goals and objectives selected when answering "where an organization wants to go". For example, if the goal is to increase profits, the customer targets are the existing customer groups [12]. On the other hand, if the goal is to increase the market growth, the customers targeted may be from a new segment [19]. Market segmentation with respect to *product's usage rates, customer/user capabilities, technology preferences, demographics,* and *purchasing power* are examples of important aspects to be considered when selecting customer targets. The choice of customer targets plays an important role in the requirements selection as the chosen customer targets set the boundaries of a product strategy.

In order to answer "how will we get there", it is important to select *primary competitive targets,* thus prioritizing competitors [11, 12].

For determining a product's position in the market it has to be differentiated based on either cost/price or value of product offering. This means that the product has to be either low priced backed by low costs or better than competitors' products as seen by customer.

The question (3) "**what to do**" addresses specific programs, "rules of the road" or tactics to be used to achieve goals and objectives established in the light of "how will we get there". This deals with the *product, pricing, promotion, distribution,* and *service* [12]. This can also be in the form of specific considerations posed by upper management. The answer also decides the selection of *strategic drivers,* whether the strategic driver is *technology-push* or *market-pull* or both.

The most important question to answer is (4) "**why would we be successful**". It needs to be clearly answered to produce a competitive product strategy [6]. The answer to this question is basically related to the differential advantage aspect of the product positioning. A solid product strategy needs to provide concrete arguments for the reason of its success in the light of customers' preferences and competitive targets. For example, if the strategy is low price, this has to be proven to be an adequate differential advantage with regards to competitors.

Roadmaps can be used to support the answering of (5) "**when will we get there**". A Roadmap is a relatively common way of representing targets based on development in the context of time and releases [15]. In this paper we highlight the role of product-technology roadmaps when it comes to defining the moving targets of strategy.

The five questions explored above (based mainly on product management literature) were used as a tentative base for the first draft of MERTS (Fig 1 – B)

3. Initial Input from Industry

In order to explore if and how requirements triage, product strategies formulation were performed and used in industry within software development organizations working in a market-driven context (Fig 1 – C) two case studies were performed. The development departments of two telecom organizations were investigated. Since the organizations preferred to remain anonymous they will be referred to as CompanyA and CompanyB.

CompanyA is a leading mobile media organization with 200,000 employees worldwide offering a convergence of communications, entertainment and information. CompanyB is leading in the area of cellular network planning and optimization with specialization in mobile network design tools having 750 employees worldwide. They have many products but ABC3G is their matured product with a large customer base.

One strategic manager and one technical manager were interviewed from each organization to capture both the strategic and technical perspectives. A constructivist perspective [20] was utilized implying that interviews were conducted with the aim of exploration and without set ideas on requirements triage and product strategy formulation practices in industry. The four subjects were interviewed separately to avoid them influencing each other. During the interviews, the interviewees did not know anything about MERTS as it was not shown to them at this stage. Below the study results are summarized:

- There was no formal method for formulating strategies, explicitly stating directions for the products. However, from the four interviews it was evident that the five strategic questions (Section 2.2) were answered implicitly while selecting requirements. The interviewees agreed that a requirement needs to be evaluated against organization's long term goals, marketing and technical concerns. The interviewees emphasized that it is important to decide where to fit a specific requirement in relation to the moving targets of the organization on the time axis. This showed that the five questions proposed in literature could be considered good-enough, and in fact are used implicitly in industry to set products' directions.
- People within the same organization did not share the same vision about a product or the organization's mission statement, primarily due to a lack of formal and explicit statements/ specifications, and the information available was not disseminated effectively.
- The requirements selection process could take 1-2 months of analysis and feasibility study due to numerous discussion sessions between strategic and technical management in the absence of formal strategies.
- Highly experienced people are pressed by time constraints and do not see writing long strategic documents as feasible. That is why they preferred to keep the strategies in their minds.

The initial input gained during the explorative interviews confirmed many of the problems described in literature.

4. MERTS

MERTS is centered on ensuring that the five strategic questions for a product are answered explicitly. Fig 2 gives an overview of MERTS and the three main parts of the method. The goal of MERTS is to offer a clear method detailing how to reach consensus and a homogenous understanding of a product strategy.

The technical product managers using the method are required to follow three parts. Each part has several steps as shown in Fig 2 (see also the exemplification.

4.1. Prerequisite of MERTS

Requirements need to be comparable to the strategies formulated. The reasoning is that MERTS is meant to assist in early requirements triage and selection. In case of requirements being too technical or too detailed method usage will not be efficient because it will be difficult to compare detailed technical requirements with strategies as strategies are formulated on a higher level of abstraction. Even if there is some process to compare detailed technical requirements with strategies they will still be too many detailed requirements to be compared against strategies. Often many detailed requirements form one product level feature/requirement therefore it is pointless to compare every detailed requirement against the strategies.

Any method for abstracting the requirements can be used, e.g. the RAM model by Gorschek and Wohlin [4], as long as it produces requirements on an abstraction level comparable to product strategies. However, the abstraction of requirements is not the focus of this paper and will not be investigated further.

4.2. Part One – Early Requirements Triage

This part provides steps to create an initial product strategy for use in requirements triage.

4.2.1. Step 1 – Specify. In order to explicitly state the goals and objectives of a product, it is important to specify the directions of movement for the product deduced from the organization’s mission statement. Thus it is important to answer the three strategic questions (**(1) Where we want to go?, (2) How to get there?, (3) What will be done?**) for each product.

The output of this step is an explicit understanding of goals and objectives associated with a specific product which can be used to perform requirements triage and selection for individual products.

To answer (1) “**Where to go**” the organization’s directions of movement have to be clearly stated. An organization can have one or many directions of movement. For example, shareholders’ revenue (as mentioned in the interviews), profit, growth, and market share (Section 2.2). The answer to this question depends on identified directions of movement and their relative importance.

The answer to (2) “**How to get there**” will bind the strategy in terms of customer segments and competition targeted and differential advantage of the individual product providing a unique selling point. In order to answer this question there is a need to specify:

- Different **customer segments targeted** by a specific product, e.g. one customer segment can be the Asian market and another can be the European market. By explicitly specifying customer segments, relative priorities can also be assigned, helping in the selection of requirements. Customer segments can be defined either on a higher level of abstraction or refined (Section 2.2) depending on the needs of the organization.
- **Competitors** to a product to show which ones to target. This enables features provided by relevant competitors to be included in the product offering. Just as customer segments, competitors can also be prioritized relatively, giving more importance to features provided by high priority competitors.

- **Differential advantage(s)** of the product that makes it unique in the market place in relation to competitors. The differential advantage can be based on any one (or combination) of technology, pricing, strategic alliances and non-functional requirements. These can also be prioritized in relation to each other depending on their importance to offering the advantage. By identifying the differential advantages and prioritizing them, it is possible to ensure that all requirements are weighted against them and requirements providing unique differential advantages are not missed.

For the answer to (3) "**What to do**" a more management centered perspective can be used, focusing on product pricing, promotion, distribution, and service (Section 2.2). However, since MERTS is targeted towards early requirements triage and selection, answers to this question will focus on the abstract technical considerations of a requirement. Some of the possible considerations rated highest by the technical experts during the interviews have been taken as example here, i.e. innovation, core assets, architecture stability, market-pull, technology-push, customization flexibility, and use of COTS [23]. Priorities can be assigned to each of these factors showing their relative importance with respect to each other.

Table 1 illustrates an example of Step 1. The columns "Questions", "Factors" and "Sub-classifications" in Table 1 show the three strategic questions and their answers.

4.2.2 Step 2 - Assign Weights. The answers from Step 1 are assigned weights. The answers of each question have a total weight of 100 (Table 1) and the total weight of the three questions is 300. The rule is to assign weights to each of the factors based on their relative importance in a way that total weight remains 100. This way has been reported to be one of easiest and quickest prioritization methods [22].

In Table 1 the answers to "Where to go?", are the factors "market growth" (assigned 60), "market share" (20), and "profit" (20). Similarly for "How to get there?", each of the three factors are assigned weights based on their relative importance, e.g. for an organization targeting growth, customer segments are more important than competitors or differential advantage, thus a higher weight is assigned to the customer segments factor compared to the competitors or differential advantage factors.

The factors mentioned in the second question's ("How to get there?") answer have sub-classifications. Similarly the targeted competitors are Comp1, Comp2, Comp3 and so on. Each of these sub-classifications is assigned weights so that sub-classifications within one factor have a total weight of 100. This has been suggested to keep the procedure straightforward and easy for the person assigning weights. Later the weights of the sub-classifications are normalized with respect to the weight assigned to the corresponding factor by multiplying the weight assigned to each of the sub-classification with the weight of the corresponding factors and then dividing by 100.

In the same way, weights are assigned to each of the factors in the last answer. The first six columns from the left in Table 1 show "Questions", "Factors", weights of each of the factors, "Sub-classifications" and weights of each of the sub-classifications, and normalization of the weights.

The number of columns used is flexible, and depends on the level of detail of a certain product strategy. This makes MERTS adaptable depending on the users (companies) preferences.

4.2.3. Step 3 - Compare Requirements. Table 1 exemplifies how a requirement is assigned points (max 100) against each factor and sub-classification based on how much the particular requirement contributes to the factor or sub-classification. The motivation being that the person performing triage (requirements comparison) does not need to worry about relative weights and normalizations. The normalization can be done automatically by the tool used (e.g. an excel sheet) after the points have been assigned against each requirement.

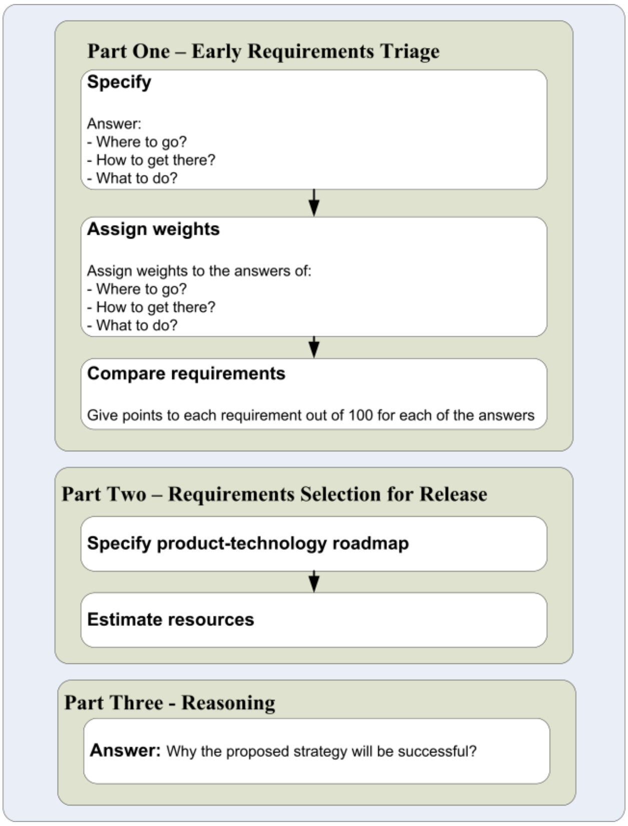

Fig 2 – MERTS steps

Requirement: *Japanese Language Support in Product XYZ*. Table 1 shows that the requirement is given 70 points out of 100 against market growth factor as it is estimated to increase growth in new markets. However, the requirement is given 0 points out of 100 for both the markets of USA and Europe as the requirement is not targeting these markets. It is given 100 out of 100 points for the Asian (Japanese) market demonstrating that this requirement is targeting the Asian market. The less the points against a factor or sub-classification the less the requirement is aligned against that particular factor or sub-classification and vice versa. As the current targeted customer segments is the Asian (Japanese) market, this requirement gets higher points when multiplied with the weight of the Asian market sub-classification.

In total the requirement got 168.4 out of 300 points when measured against different strategic factors and sub-classifications. By running this step for all requirements, points for all the requirements can be obtained and a threshold can be defined. Requirements with points above this threshold get selected.

The first three steps of MERTS should be performed at product management level supporting the triage of requirements (aiding in the selection). The purpose of step 3 (Section 4.2.3) is not requirements prioritization which is usually associated with early project activities during release planning. The points assigned to each requirement, against each factor or sub-classification, show the level of strategic alignment.

4.3. Part Two – Requirements Selection for Release

After a set of requirements (deemed to be aligned with the strategy) have been selected, the question in focus is "when to get there". This activity consists of two steps described in sections 4.3.1 and 4.3.2

4.3.1. Step 1 - Specify product-technology roadmap. It has been emphasized in literature [15] to chalk out a product-technology roadmap to get an overview of the relationship between product releases (product evolvement) and successive technology generations. This means specifying what a product tends to achieve along the time axis in term of its evolvement and technology trends. This enables placement of requirements in appropriate intervals planned in a roadmap. For example, if a requirement requires expertise in some new technology to be explored in the future and this has been planned in the roadmap, the requirement can be postponed or depending on the urgency of the requirement, the roadmap can be altered.

4.3.2. Step 2 - Estimate resources.
In order to determine the feasibility of the requirements, the organization needs to explicitly state financial and effort allowances against each interval in the roadmap. Several methods can be used to estimate cost, effort and time, e.g. feature points, function points, lines of code, and methods like e.g. COCOMO [19] can be used to support the efforts. An alternative could be to perform estimates based on previous development efforts. Additionally, requirements prioritization techniques [22] can be used to plan releases for the product.

4.4. Part Three – Strategy Rationale

Once the strategic questions have been answered, it is important to document the reasoning behind the decisions. This way if the decisions (and indirectly the answers) result in success (of a product) replication can be achieved, and the organization has good examples to follow for future efforts.

In addition, the strategy formulated through MERTS should be used to share product and organizational visions across the organization. In its simplest form it can mean writing a paragraph explaining the reason behind the answers, keeping in view the organization's long term goals, financial plans, technology trends and marketing trends.

5. Industrial Validation of MERTS

After the creation of MERTS an initial validation of the method was performed in industry utilizing the same resources as described in the exploratory study (Section 3). The company representatives tested the method and were asked to render their opinions from different aspects listed below.

Practicality and Applicability of the Method. All the four interviewees were of the opinion that the method is practical and easy to use once it is clearly understood. They agreed that the method is customizable based on the organization’s individual goals and objectives. All four interviewees showed interest in implementing the method. In fact, CompanyB mentioned that they would be considering MERTS for helping them with requirements triage for their next product.

Method Implementation. The prerequisite (Section 4.1) for implementing this method was not considered as an overhead. This is because the interviewees realized the need of requirements abstraction for early triage and selection.

From the method implementation perspective, it was agreed that the method provides a starting point to formalizing and documenting aspects of the organization’s directions of movement, i.e. strategies. It was agreed that it is very important to think and align requirements with the organization’s product strategies. The existing processes in the companies did not achieve this, thus MERTS was considered a pragmatic start that required a reasonable effort investment.

It was suggested that technical product managers assign the weights because currently mostly they do this already although ad-hoc without using MERTS.

The method would offer decision support and structure, as well as clarifying business goals, objectives, and enable requirements triage. However, it was also recognized that it would be a time consuming and resource intensive process to gather different aspects of the business goals from different stakeholders and formalizing them to a common agreed upon set. Similarly, assigning weights to selected factors would require substantial discussion among stakeholders. But the interviewees agreed that once the strategies were made available in the suggested format, the time to select requirements would definitely decrease, in addition to making the choices more aligned with the goals. Moreover, the interviewees liked the method as it could provide a common understanding of strategies and organizational direction thus reducing communication gaps, especially between managers and technical experts.

Step by step requirements selection, resources estimation and then road mapping were thought to be logical steps for requirements realization.

Table 1 – Example demonstrating MERTS steps

Questions	Factors	Weightings of Factors	Sub classification	Sub classifications Weightings	Normalized Weightings	Requirement (Japanese language support)	Normalized
Where	Market growth	60			60	70	42
	Market share	20			20	70	14
	Profit	20			20	50	10
		100			***100***		
How	Customer segments	40	*USA market*	20	8	0	0
			European market	30	12	0	0
			Asian market	50	20	100	20
				100			
	Competitors	30	*ABC company*	70	21	100	21
			HIJ company	20	6	0	0
			Others	10	3	0	0
				100			
	Differential advantage	30	*Innovation*	60	18	70	12.6
			Pricing	5	1.5	0	0
			Strategic Alliances	10	3	0	0
			Non functional requirements	25	7.5	0	0
		100		***100***	***100***		
What	New Technology	40			40	80	32
	Use of core assets	11			11	0	0
	Architecture stability	20			20	40	8
	Market pull	10			10	80	8
	Technology push	15			15	0	0
	Customization flexibility	2			2	0	0
	COTS	2			2	40	0.8
		100			***100***	**Total**	**168.4**

6. Conclusion

This paper presents MERTS, developed using well established concepts for formulating strategies inspired by good practices in management literature. In addition a set of interviews were conducted with industry managers and technical experts in two organizations to explore important factors affecting the formulation of strategies from an industrial perspective. The factors affecting requirements triage and selection were also explored. With these factors identified, it was concluded that a gap between strategic managers' and technical experts' perspectives existed, as well as there was lack of a structured approach to agree on goals and the ability to perform requirements triage following these goals. MERTS enables professionals to work with organizational directions of movement (strategies) in a structured manner. The benefits are many as described and confirmed during initial validation (Section 5).

It has to be realized though that the use of MERTS requires an investment of time and resources to formalize the strategies and homogenize understanding. Although, the investment is not continuous but mostly initial, and should be alleviated by the improved triage, not to mention the benefit of having goals explicitly shared and understood in the organization.

During the creation of MERTS we did not adopt a one-size-fit-all philosophy with regards to strategies, but rather by modifying the factors chosen, a “tailoring” towards a product (or organization) can be achieved prior to use in order to adapt to organizational and product specific goals and objectives. This makes it possible for any organization to use MERTS to the degree needed, making it suitable for larger companies as well as small and medium sized enterprises. The contribution of MERTS is to support explicit discussions, and the formulation and documentation of strategies, enabling requirements triage.

Feedback regarding MERTS was obtained, but it was limited in time and scope, thus can be described as an artificial environment from the construct validity perspective. Using the same subjects for validation and initial industry input could result in a bias towards MERTS. This risk was alleviated through the fact that MERTS was not shown to the subjects.

At present the validation of MERTS has been limited in scope. Future work involves additional validation, both static through interviews and brainstorming with industry professionals, but also dynamic validation in the form of industry piloting. In addition it is important to validate the method against additional domains (in addition to the telecom domain). Further validation will investigate issues such as scalability, usability and usefulness.

With the help of further static and later dynamic validations, efforts will be made to reduce the subjectivity in steps “Specify” (Section 4.2.1) and “Assign weights” (Section 4.2.2). Moreover, MERTS has to be refined to take requirements dependencies into consideration during triage.

7. References

[1] E. Carmel and S. Becker, "A process model for packaged software development," *IEEE Transactions on Engineering Management*, vol. 42, pp. 50-62, 1995.

[2] K. El Emam and N. H. Madhavji, "A field study of requirements engineering practices in information systems development," in *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, 1995, pp. 68-80.

[3] K. Lena, G. D. Åsa, J. N. o. Dag, R. Björn, and P. Anne, "Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study," in *Eighth International Workshop on Requirements Engineering*, Germany, 2002, pp. PP 101-112. [4] T. Gorschek and C. Wohlin, "Requirements Abstraction Model," *Requirements Engineering*, vol. 11, pp. 79-101, 2006.

[5] C. Potts, "Invented requirements and imagined customers: requirements engineering for off-the-shelf software," in *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, , pp. 128-130, 1995.

[6] McGrath and E. Michael, *Product Strategy for High Technology Companies*, 2nd edition ed.: McGraw-Hill Professional Book Group, 2000.

[7] M. Crawford and A. D. Benedetto, *New Products Management*, 7 edition ed.: McGraw-Hill/Irwin, 2002.

[8] A. Aurum and C. Wohlin, "Aligning Requirements with Business Objectives: A Framework for Requirements Engineering Decisions," in *Proceedings of Requirements Engineering Decision Support Workshop*, Paris, France, 2005.

[9] J. Vähäniitty, "Product Strategy Decisions in Small Software Product Businesses – A Framework and Experiences," in *Institute of Software Business and Engineering*. Master thesis, Helsinki University of Technology, 2003.
[10] A. M. Davis, "The art of requirements triage," *IEEE Computer,* vol. 36, pp. 42-49, 2003.
[11] J. M. Utterback, *Mastering the Dynamics of Innovation*: Boston: Harvard Business School Press, 1994.
[12] D. Lehmann and R. Winer. " *Product Management*, 3rd edition ed.: McGraw-Hill Irwin, 2002.
[13] W. Oliver Richard, "Real-time strategy: What is strategy, anyway?," *The Journal of Business Strategy,* vol. 22, pp. 7-10, 2001.
[14] E. Simmons, "Requirements Triage: What Can We Learn from a "Medical" Approach?," *IEEE Software,* vol. 21, pp. 86-88, 2004.
[15] A. Kappel Thomas, "Perspectives on road maps: How organizations talk about the future," *The Journal of Product Innovation Management,* vol. 18, pp. 39-50, 2001.
[16] T. Gorschek, and A. Davis, "Requirements Engineering: In Search of the Dependent Variables", *Accepted in Information and Software Technology* (can be obtained by email) N/A, (N/A), 2006, N/A
[17] M. Scott George, "Top priority management concerns about new product development," *The Academy of Management Executive,* vol. 13, pp. 77-84, 1999.
[18] V. Krishnan and T. U. Karl, "Product Development Decisions: A Review of the Literature," *Management Science,* vol. 47, pp. 1-21, 2001.
[19] N. E. Fenton and S. L. Pfleeger, *Software Metrics - A Rigorous & Practical Approach*, 2nd Edition ed.: International Thomson Computer Press, 1996.
[20] J. W. Creswell, *Research Design - Qualitative, Quantitative and Mixed Method Approaches*, 2nd Edition ed.: Sage Publications, 2002.
[21] J. Vähäniitty, "Key decisions in strategic new product development for small software product businesses," in *Proceedings of 29th Euromicro Conference*, 2003, pp. 375-383.
[22] P. Berander, "Evolving Prioritization for Software Product Management," in APS, PhD thesis, Ronneby: Bleking Institute of Technology, 2007, p. 250.
[23] K. Mahvish and A. Khurum, "A Model for Early Requirements Triage and Selection Utilizing Product Strategies," in APS. Master thesis 2007:01, Blekinge Institute of Technology,2007,http://www.bth.se/fou/cuppsats.nsf/1d345136c12b9a52c1256608004f0519/986067aa8859ae27c125726d00326120!OpenDocumenthttp://www.bth.se/fou/cuppsats.nsf/1d345136c12b9a52c1256608004f0519/986067aa8859ae27c125726d00326120!OpenDocument

Chapter 5 - A Controlled Experiment of a Method for Early Requirements Triage Utilizing Product Strategies

Mahvish Khurum[1], Tony Gorschek[1], Lefteris Angelis[2], Robert Feldt[1]

[1]Blekinge Institute of Technology, Department of Systems and Software Engineering, S-372 25, Ronneby, Sweden
mkm@bth.se, tgo@bth.se, rfd@bth.se

[2]Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
lef@csd.auth.gr

Abstract. [Context and motivation] In market-driven product development of software intensive products large numbers of requirements threaten to overload the development organization. It is critical for product management to select the requirements aligned with the overall business goals, product strategies and discard others as early as possible. Thus, there is a need for an effective and efficient method that deals with this challenge and supports product managers in the continuous effort of early requirements triage [1, 2] based on product strategies. This paper evaluates such a method – A Method for Early Requirements Triage Utilizing Product Strategies (MERTS), which is built based on the needs identified in literature and industry. **[Question/problem]** The research question answered in this paper is "If two groups of subjects have a product strategy, one group in NL format and one in MERTS format, will there be a difference between the two groups with regards to effectiveness and efficiency of requirements triage?" The effectiveness and efficiency of the MERTS were evaluated through controlled experiment in a lab environment with 50 software engineering graduate students as subjects. **[Principal ideas/results]** It was found through results that MERTS method is highly effective and efficient. **[Contribution]** The contribution of this paper is validation of effectiveness and efficiency of the product strategies created through MERTS method for requirements triage, prior to industry trials. A major limitation of the results is that the experiment was performed with the graduate students and not the product managers. However, the results showed that MERTS is ready for industry trials.

Keywords: Market driven requirements engineering, requirements triage, product strategies, MERTS, experiment, effectiveness and efficiency.

1 Introduction

Due to the emergence of markets for off-the-shelf or packaged software [3, 4], market-driven development is gaining increased interest in comparison to customer-specific system development [5, 6]. As a consequence a shift in focus is occurring, affecting software development in general and requirements engineering in particular [6]. In contrast to traditional requirements engineering, requirements in market-driven requirements engineering (MDRE) to a large extent come from internal sources such as developers, marketing, sales teams, support, bug reports, as well as from external sources such as different users, customers from different and multiple market segments, and competitors [7]. The result is a large and continuous flow of requirements that threaten to overload the development organization [5]. This has two major implications. One, the product and domain knowledge reside largely with the development company itself. For example a developer of robotics products with many of e.g. car manufacturers as customers probably knows more about robotics than any one customer.

Two, the risk and cost of development is carried by the development organization, meaning that the potential revenues depend on selecting the “right” requirements for implementation. The selection accuracy is the main success criteria for the development organization, and being able to perform the selection in a scalable and cost effective way is crucial to avoid overloading. Which requirements to select is a trade-off between different strategic factors such as key-customer requirements and long-term aspects and innovation efforts. All of these factors, and more, need to be explicitly stated and weighed together to reach an optimal strategy for the company, which can then be used for selecting the “right” requirements for implementation.

However, while industry managers regard strategy formulation and use as the most important aspect of technology management [8], strategy formulation is usually performed ad-hoc, and a systematic approach for formulating strategies is often missing in practice [9]. Even if the formulation of strategies was pursued, the factors affecting strategy formulation differ between different stakeholders. Strategic and middle management and technical experts all need to share one vision. Strategic managers often overlook the technical perspective, and technical experts can be unaware of or overlook the strategic managers’ perspective. As a result of these challenges, identified both in academia and through industry case studies, a Method for Early Requirements Triage and Selection (MERTS) [10] was developed to combine both strategic and technical perspectives for the formulation of product strategies that are good-enough to be used for early requirements triage and selection.

This paper presents an experiment testing some key aspects of this method, following a stepwise plan to validate MERTS prior to industry piloting.

MERTS has two main purposes. First, it acts as a stepwise guide to creating product strategies taking both strategic and technical views into account thus following a systematic way of agreeing on a joint plan. Secondly, the strategies resulting from MERTS can be used by product managers to effectively perform requirements triage and requirements selection in a reasonable amount of time as spending initial 10 minutes on triage versus 10 hours is super critical for industry. The experiment aims at testing the second purpose of MERTS. Thus, the main purpose of the experiment is to assess the efficiency and effectiveness of requirements triage utilizing strategy formulated and formatted using MERTS prior to industry piloting. Thus, this experiment is considered as a lab validation following the research approach suggested by Gorschek et al. [11] aimed at producing useable and useful research results and successful technology transfer.

Before describing the experiment and experiment results an introduction to MERTS is given in Section 2. Section 3 details the experiment design. Section 4 lists the validity threats. Section 5 contains preparation and execution details. Section 6 presents the results and analysis, and finally Section 7 presents the conclusions drawn and plans for further work.

2 MERTS Background

MERTS is centered on ensuring that the five strategic questions for a product are answered explicitly [10]. Fig. 1 gives an overview of MERTS and the three main parts of the method. The goal of MERTS is to offer a clear method detailing how to reach consensus and a homogenous understanding of a product strategy. The product managers using the method are required to follow these three parts. Each part has several steps (see Fig. 1).

Part One – Early Requirements Triage. This part provides steps to create an initial product strategy for use in requirements triage.

A. Specify. In order to explicitly state the goals and objectives of a product, it is important to specify the directions of movement for the product deduced from the organization's mission statement. Thus it is important to answer the three strategic questions ((1) Where we want to go?, (2) How to get there?, (3) What will be done?) for each product.

The output of this step is an explicit understanding of goals and objectives associated with a specific product which can be used to perform requirements triage and selection for individual products.

To answer (A.1) "**Where to go**" the organization's directions of movement have to be clearly stated. An organization can have one or many directions of movement. For example, shareholders' revenue, profit, growth, and market share [10]. The answer to this question depends on identified directions of movement and their relative importance.

The answer to (A.2) "**How to get there**" will bind the strategy in terms of customer segments and competition targeted and differential advantage of the individual product providing a unique selling point. In order to answer this question there is a need to specify:

- Different customer segments targeted by a specific product, e.g. one customer segment can be the Asian market and another can be the European market. By explicitly specifying customer segments, relative priorities can also be assigned, helping in the selection of requirements. Customer segments can be defined either on a higher level of abstraction or refined depending on the needs of the organization.
- Competitors to a product to show which ones to target. This enables features provided by relevant competitors to be included in the product offering. Just as customer segments, competitors can also be prioritized relatively, giving more importance to features provided by high priority competitors.
- Differential advantage(s) of the product that makes it unique in the market place in relation to competitors. The differential advantage can be based on any one (or combination) of technology, pricing, strategic alliances and non-functional requirements. These can also be prioritized in relation to each other depending on their importance to offering the advantage. By identifying the differential advantages and prioritizing them, it is possible to ensure that all requirements are weighted against them and requirements providing unique differential advantages are not missed.

For the answer to (A.3) "**What to do**" a more management centered perspective can be used, focusing on product pricing, promotion, distribution, and service. However, since MERTS is targeted towards early requirements triage and selection, answers to this question will focus on the abstract technical considerations of a requirement. Some of the possible considerations rated highest by the technical experts during the interviews have been taken as example here, i.e. innovation, core assets, architecture stability, market-pull, technology-push, customization flexibility, and use of COTS [10]. Priorities can be assigned to each of these factors showing their relative importance with respect to each other.

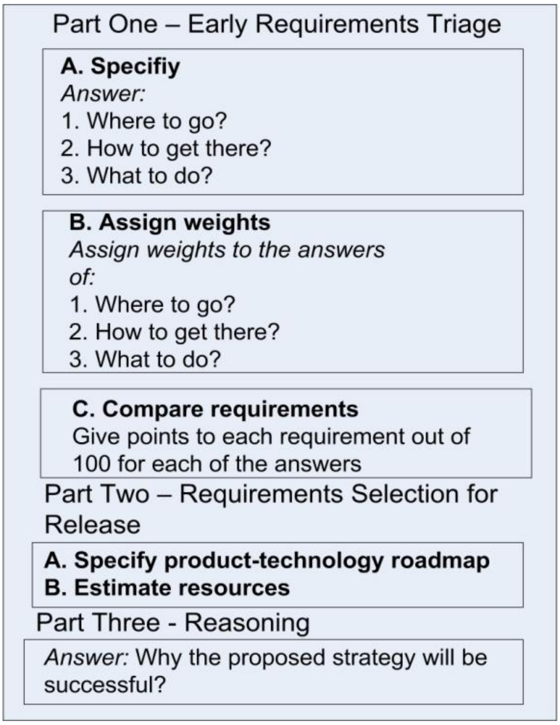

Fig. 1. MERTS Steps

B. Assign Weights. The answers from Step 1 are assigned weights. The rule is to assign weights to each of the factors based on their relative importance in a way that total weight remains 100. This way has been reported to be one of easiest and quickest prioritization methods [17].

C. Compare Requirements. The total weights of all the requirements are compared against a threshold to select or reject each of the requirements.

The first three steps of MERTS should be performed at product management level supporting the triage of requirements (aiding in the selection). The purpose of step 2 (Assign weights) is not requirements prioritization which is usually associated with early project activities during release planning. The points assigned to each requirement, against each factor or sub-classification, show the level of strategic alignment.

Part Two – Requirements Selection for Release. After a set of requirements (deemed to be aligned with the strategy) have been selected, the question in focus is "when to get there". To answer this following two steps are required.

A. Specify product-technology roadmap. It has been emphasized in literature [12] to chalk out a product-technology roadmap to get an overview of the relationship between product releases (product evolvement) and successive technology generations. This means specifying what a product tends to achieve along the time axis in term of its evolvement and technology trends. This enables placement of requirements in appropriate intervals planned in a roadmap.

For example, if a requirement requires expertise in some new technology to be explored in the future and this has been planned in the roadmap, the requirement can be postponed or depending on the urgency of the requirement, the roadmap can be altered.

B. Estimate resources. In order to determine the feasibility of the requirements, the organization needs to explicitly state financial and effort allowances against each interval in the roadmap. Several methods can be used to estimate cost, effort and time, e.g. feature points, function points, lines of code, and methods like e.g. COCOMO [13] can be used to support the efforts. An alternative could be to perform estimates based on previous development efforts. Additionally, requirements prioritization techniques [14] can be used to plan releases for the product.

Part Three – Strategy Rationale. Once the strategic questions have been answered, it is important to document the reasoning behind the decisions. This way if the decisions (and indirectly the answers) result in success (of a product) replication can be achieved, and the organization has good examples to follow for future efforts.

In addition, the strategy formulated through MERTS should be used to share product and organizational visions across the organization. In its simplest form it can mean writing a paragraph explaining the reason behind the answers, keeping in view the organization's long term goals, financial plans, technology trends and marketing trends.

In order to implement MERTS method, requirements need to be comparable to the strategies formulated. The reasoning is that MERTS is meant to assist in early requirements triage and selection. In case of requirements being too technical or too detailed method usage will not be efficient because it will be difficult to compare detailed technical requirements with strategies as strategies are formulated on a higher level of abstraction. Even if there is some process to compare detailed technical requirements with strategies they will still be too many detailed requirements to be compared against strategies. Often many detailed requirements form one product level feature/requirement therefore it is pointless to compare every detailed requirement against the strategies. Any method for abstracting the requirements can be used, e.g. the RAM model by Gorschek and Wohlin [6], as long as it produces requirements on an abstraction level comparable to product strategies.

3 Design of the Controlled Experiment

The usefulness and the usability of MERTS depend on several concepts that need to be assessed; the one studied in this controlled experiment is that it should be possible to perform triage of a new incoming requirement based on its alignment with the MERTS strategy. This means that a MERTS strategy should be usable in an efficient and effective manner for performing requirements triage (formally stated as research questions in Section 3.3).

3.1 Context

The experiment was conducted in an academic setting, with the help of 50 engineering graduate students at Blekinge Institute of Technology. It was conducted as a mandatory although non-graded exercise at the end of a 7.5 ECTS merits master's course in research methodology. Participation was mandatory and despite the ethical issues of forcing subjects to participate in a study, it was believed that the experiment had several pedagogical benefits in the course. The students were instead given the option to exclude their individual results from the study, an option not utilized by any student. The intended users of MERTS, however, are product managers with several years of experience in a specific domain and product. In the experiment, the subjects have no training in using MERTS, they possess limited domain knowledge, are under time pressure, and most of them have not seen the product strategies or the requirements before. There is thus a considerable gap between the intended target group and the sample used in this experiment. The subjects in this experiment can be expected to adapt a more surface oriented approach to the problem than product managers. We argue that this works to our advantage, since any results that we evaluate are likely to stem from the instrumentation and the use of MERTS, rather than previous experiences of the subjects in the study. If MERTS proves to be usable in the experiment, it would indicate that it is able to decrease the dependency on individual's experience, product knowledge, and methodology.

3.2 Subjects

The group of experiment subjects using MERTS strategy for requirements triage had an average of 1.46 years of industrial experience, and only 3 out of 25 subjects had seen project strategies or performed requirements triage before. The subjects in the group using a natural language (NL) strategy for requirements triage had an average experience of 2.23 years, and 4 subjects out of 25 had seen product strategies in some form in their companies and 5 had performed requirements triage previously. This information was gathered through a post-experiment questionnaire; the groups were formed randomly.

3.3 Research Questions

The main research question is RQ which is described below along with associated hypotheses and independent/dependant variables.

RQ**:** If two groups of subjects have a product strategy, one group in NL format and one in MERTS format, will there be a difference between the two groups with regards to effectiveness and efficiency?

Hypotheses

***Null hypothesis, $H_{o\ Effectiveness}$*:** The use of MERTS strategy for requirements triage is not significantly different from NL strategy with regards to effectiveness.

***Alternative hypothesis, $H_{1\ Effectiveness}$*:** The use of MERTS strategy for requirements triage is significantly different from NL strategy with regards to effectiveness.

***Null hypothesis, $H_{o\ Efficiency}$*:** The use of MERTS strategy for requirements triage is not significantly different from NL strategy with regards to efficiency.

***Alternative hypothesis, $H_{1\ Efficiency}$*:** The use of MERTS strategy for requirements triage is significantly different from NL strategy with regards to efficiency.

Variables Selection: This experiment has the following independent variables:

Independent variables Product strategy formatted according to MERTS or according to NL.
Dependant variables The dependant variables are *effectiveness* and *efficiency* measured through: 1. Effectiveness: Number of correct requirements triage decisions. 2. Efficiency: Time taken (in minutes) to perform triage on all requirements.

The definition and hypotheses for finding an answer to RQ depict that the design is: *one factor with two treatments*. The factor is the product strategy and treatments are NL and MERTS.

3.4 Design and Instrumentation

Prior to the experiment execution one round of validation (pre-test) was performed to refine the experiment design and instrumentation. The pre-test was constructed and evaluated with the help of 4 colleagues at Blekinge Institute of Technology. Based on the experience from this pre-test, the experiment package was revised. Specifically, the initial presentation describing the study and running through an example was refined to make it more concrete for describing the motivation of the triage decisions taken by the subjects.

The subjects were divided randomly into two groups, with one treatment per group. The experiment consisted of two parts that both ran consecutively without breaks. The first part was a preparatory lecture where the concepts of requirements triage and MERTS/NL were introduced, together with a presentation of the experiment, research instruments and an example of how to take triage decisions and provide motivations. The second part of the experiment consisted of filling the forms. All other artifacts like the requirements and forms were the same.

During the experiment, the following instruments were used:

- Each subject was given an example of how to perform triage using either NL or MERTS (depending on the group).
- Each subject was given either NL or MERTS formatted strategy for the experiment. The product strategy detailed the goals of a new version of a mobile phone targeted for entertainment-oriented users in the Asian market. The level of information in the two strategies was the same with respect to goals and objectives, targeted customers and competitors, differential advantages and technical considerations. NL strategy was formulated based on example strategies given in literature. Industrial experience of authors with real product strategies was also beneficial to ensure that NL formatted strategy was as close as possible to industry practice. The MERTS strategy however, as prescribed by the MERTS method, had weights assigned to each of the factors stated in the strategy which was absent in NL strategy because in traditional NL strategies the weights to each of the factors is not explicitly given in numbers rather stated as subjective statements.
- The requirements set contained 13 product and 18 feature level requirements. For example, messages communication, music playing, enhanced imagining, enhanced display, availability, usability, browsing, connectivity, and so on. The requirements were constructed to be of moderate to good quality based on industry observation. The appropriateness of the requirements and other instruments was also validated in the pre-test. It is important to understand that in lab experimentation, it is not possible to have a large number of requirements for triage. There is a limited amount of time where subjects have to understand the method and then apply it for requirements triage. The aspects of effectiveness and efficiency as evaluated in the experiment are however related to using MERTS strategies vs. NL strategies. The relative efficiency and effectiveness is the goal.
- Each requirement in the set has at least two levels: product and feature, and often also divided into functions. Each requirement was formatted and specified using the following attributes; Unique Id, Product level requirement, Feature level requirement, Function level requirement, Component level requirement (in some cases) and Comments.
- The instrumentation had a *Decision* column next to every feature level requirement with two options: Accept and Reject. For every triage decision the experiment subject had to specify a

rationale behind the triage (Accept or Reject) decision. It was emphasized during the experiment training that the motivation had to be deduced from the product strategy and not personal judgments and opinions. Last in the experiment, each subject had to answer the questions at the end of experiment as a post-test. The experiment materials (NL strategy, MERTS strategy, example requirements and the post-test) is not included in the paper as space does not allow, but can be obtained online at http://www.bth.se/tek/aps/mkm.nsf/pages/merts-experimentation.

4 Validity Evaluation

Internal validity. This threat can have a huge impact on the experiment results if the data collection forms and other instruments are poorly designed. To ensure that the research instruments, including the posed question, are of a good quality, one pre-test with the material was executed before the "live" round. Moreover, all subjects received the same introductory lecture, and were given the same material in the same order. It is thus unlikely that the instrumentation and the implementation of the treatment influenced the results unduly. That being said, since we used the answers of human subjects as measures, the gathered measures are of course not 100% repeatable.

To alleviate author's bias towards MERTS while designing the experiment, a senior researcher (the second author) not involved in the creation of MERTS, was actively involved in the design of the experiment to avoid favoritism towards MERTS.

Construct validity. To reduce the risk of evaluation apprehension among the test subjects, they were told that they would be graded on their efforts, but not on the number of correct decisions.

External validity. To ensure the external validity and the ability to generalize the results, we use a requirements specification from a fairly mature domain. As discussed in Section 3.2, the knowledge and experience of the participants is less than that of the target audience (e.g. product managers in industry). To reduce this gap, a system from a domain that is familiar to the subjects was used.

The correlation analysis between the total number of correct triage decisions and the industrial experience show that there was no significant difference between performance of subjects with more industry experience and those with less experience (both for the group using MERTS strategy and group using NL strategy). Thus, the two groups were homogenous in terms of industry experience.

As the intended target of MERTS (e.g. product managers) would have not only a better requirement and domain understanding, but also more experience in triage, it can be argued that the ability to use MERTS (and the potential positive results obtained in the experiment) should be transferrable to industry practice. Moreover, experimentation using state-of-the-art research (well-structured method MERTS in this case) also has learning/training benefits for future professionals.

In this study paper printouts were used, which may impact the readability and the ease by which the participants may access the information. Hence, any positive effects are also here transferable to the target audience and the target environment as the use of tools may increase usability.

5 Operation

The subjects were not aware of the aspects intended for study, and were not given any information regarding research questions in advance. They were aware of the fact that it was a controlled experiment in the area of requirements engineering that was a part of their research methodology course. The experiment ran over a period of three hours, and the subjects were randomly divided into two groups seated in two different rooms. Introduction to the experiment was given during these three hours in the form of a brief slide show presentation. In this presentation basic concepts of product strategy and requirements triage were explained along with examples.

The mean time to conduct the experiment was around 60 minutes when using MERTS strategy, the shortest time spent was around 33 minutes and the longest was 107 minutes. The group using NL strategy had a mean time of around 33 minutes, the shortest time spent was 17 minutes and the longest was 50 minutes.

6 Results and Analysis

6.1 Testing $H_{o\ Effectiveness}$

In each group 18 feature level requirements were given to the subjects and they had to decide which of these are to be selected/rejected in accordance with the product strategy (either MERTS or NL). According to the experiment design 10 feature level requirements were to be selected and 8 rejected based on the product strategies. During this analysis, answers that were in line with the study design and aptly motivated were treated as "correct". If an answer is in line with the study design but missing a proper motivation (that is the motivation is not based on the given product strategy) or if the answer is not in line with the study design, the answer is considered "incorrect".

Table 1 shows the mean, standard deviation, skewness and kurtosis for the total number of correct decisions for all the 18 feature level requirements for the two strategies: MERTS and NL respectively. The results show that the average number of correct decisions using MERTS (Mean = 17.72) is more than double the average number of correct decisions using the NL (Mean = 6.22).

Table 1. Statistics for total number of correct decisions for MERTS and NL strategies

MERTS		**Natural Language**	
Statistic	**Value**	**Statistic**	**Value**
Mean	17.72	Mean	6.22
Median	17.50	Median	5.00
Std. deviation	4.456	Std. deviation	4.124
Skewness	-.143	Skewness	1.180
Kurtosis	-1.257	Kurtosis	0.639

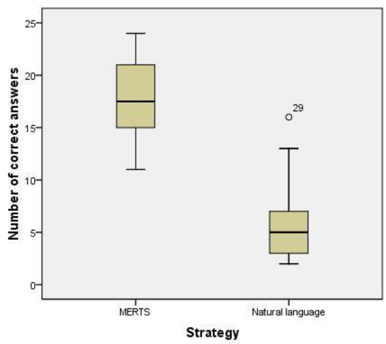

Fig. 2. Boxplots for total number of correct decisions using the two strategies

Confirmed complementary view is offered by the boxplots in Fig. 2 where the greater number of correct triage decisions using MERTS strategy is evident. Through the boxplots, an outlier (marked with a small circle and 29) was identified which is discussed below.

The skewness and kurtosis values for the total number of correct triage decisions show that the distributions seem to differ from the normal distribution. To check normalization prior to the application of an appropriate statistical test, normality tests were performed on the given data and the results are shown in Table 2. It can been in Table 2 that the total number of correct triage decisions for MERTS do not differ significantly from the normal distribution (Significance = 0.20 > 0.05) but the distribution of the total number of correct triage decisions for NL is not normally distributed (Significance = 0.048 < 0.05). Based on this result the Mann-Whitney U test

Table 2. Normality tests for total number of correct decisions.

Strategy	**Kolmogorov-Smirnov**		**Shapiro-Wilk**	
	Statistic	Sig.	Statistic	Sig.
MERTS	0.158	0.200	*0.923*	0.146
NL	0.203	0.048	0.862	0.013

was used to compare if the two sample distributions (of total number of correct decisions using MERTS and NL strategies) come from the same population.

Looking at overall effectiveness of MERTS versus NL strategy the bar chart in Fig. 3 confirms that MERTS was more effective for triage decisions than NL.

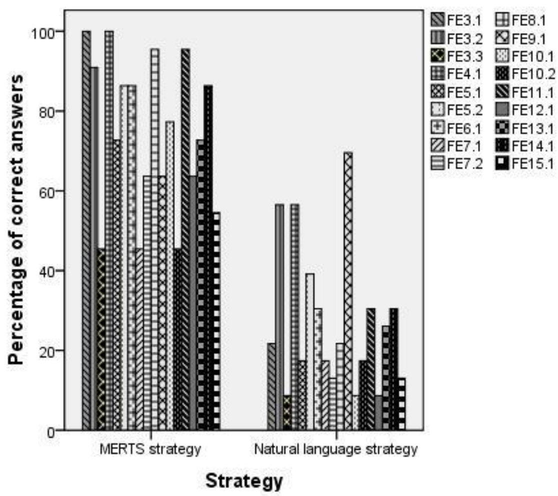

Fig. 3. Percentage of correct decisions in relation to strategy used.

Additionally, the visual inspection of cross tabulations in Table 3 shows that the percentages of correct decisions for MERTS are significantly higher than the correct decisions for NL. For example, looking at second row it is possible to see that 22 subjects (62.9%) using MERTS strategy made a correct triage decisions for FE3.2 whereas only 13 subjects (37.1%) using NL strategy made a correct decisions. The difference of the percentages of correct and incorrect decisions between the two groups was tested with the chi-square test. For all the requirements the difference is significant at 0.025 except the requirement 9.1. This is the same requirement for which an outlier has been identified in Fig. 2. The reason behind this is that for this particular requirement the total number of correct decisions both for MERTS and NL are equal (16 correct decisions).

Table 3. Significance using Chi-Square Test.

Requirement	**Sig.**	**MERTS**				**Natural Language**			
		Incorrect		***Correct***		***Incorrect***		***Correct***	
		Count	**%**	**Count**	**%**	**Count**	**%**	**Count**	**%**
FE3.1	0.000	0	0	25	83.3	20	100	5	16.7
FE3.2	0.002	2	14.3	22	62.9	12	85.7	13	37.1
FE3.3	0.003	13	36.1	11	84.6	23	63.9	2	15.4
FE4.1	0.000	0	0	25	65.8	12	100	13	34.2
FE5.1	0.000	7	25	17	81	21	75	4	19
FE5.2	0.000	3	15.8	21	70	16	84.2	9	30
FE6.1	0.000	3	14.3	22	75.9	18	85.7	7	24.1
FE7.1	0.024	13	38.2	11	73.3	21	61.8	4	26.7
FE7.2	0.000	9	29	15	83.3	22	71	3	16.7
FE8.1	0.000	1	4.8	24	82.8	20	95.2	5	17.2
FE9.1	1.000	9	50	16	50	9	50	16	50
FE10.1	0.000	6	22.2	18	90	21	77.8	2	10
FE10.2	0.047	14	42.4	11	73.3	19	57.6	4	26.7
FE11.1	0.000	1	5.3	23	26.7	18	94.7	7	23.3
FE12.1	0.000	10	31.2	15	88.2	22	68.8	2	11.8
FE13.1	0.002	8	29.6	17	73.9	19	70.4	6	26.1
FE14.1	0.000	3	15	22	75.9	17	85	7	24.1
FE15.1	0.001	11	33.3	14	82.4	22	66.7	3	17.6

Requirement 9.1 is shown below with its related Function and Product level requirements. This requirement is easy to relate to in both strategy formulations, and also during the requirements engineering course at the university, students were given an example of a very similar requirement. In perfect hindsight it was not surprising that 50% subjects got this requirement decision correct both in MERTS strategy and NL strategy. The conclusion drawn after analysis was that the use of this particular requirement in the experiment was less than optimal.

Requirement 9.1.
Product: PR9: Usability Internationally **Feature: FE9.1: The mobile shall support multiple languages.** Function: FN9.1.1: The mobile shall provide Swedish language support FN9.1.2: The mobile shall provide Chinese language support FN9.1.3: The mobile shall provide Japanese language support.

Finally to confirm the results, the Mann-Whitney U test is applied in order to check the significance of the results. Significance less than 0.001 was attained, indicating that there is a significant difference between the means of the two groups. The null hypothesis: $H_{oEffectiveness}$ is rejected and $H_{1Effectiveness}$ is confirmed, i.e. using MERTS is significantly different from NL for requirements triage with regards to effectiveness. To conclude, the use of MERTS strategy for requirements triage is superior to NL strategy with regards to effectiveness.

6.2 Testing $H_{0\ Efficiency}$

Fig. 4 shows the mean, standard deviation, skewness and kurtosis values for the time taken by the 50 subjects using the MERTS and NL strategies. The results show that average time taken using MERTS (Mean = 60.12) is double the average time taken using NL (Mean = 33.44). The outlier identified in Fig. 4 contributes to the large mean and standard deviation for the triage time taken using MERTS (Std. Deviation 19.10).

Table 4. Total time taken (minutes) for MERTS and NL strategies.

MERTS		Natural Language	
Statistic	**Value**	**Statistic**	**Value**
Mean	60.12	Mean	33.44
Median	59.00	Median	34.00
Std. deviation	19.10	Std. deviation	9.10
Skewness	0.93	Skewness	-0.06
Kurtosis	0.62	Kurtosis	-0.92

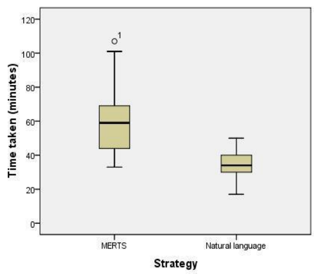

Fig. 4. Boxplots for time taken (minutes) for the two strategies.

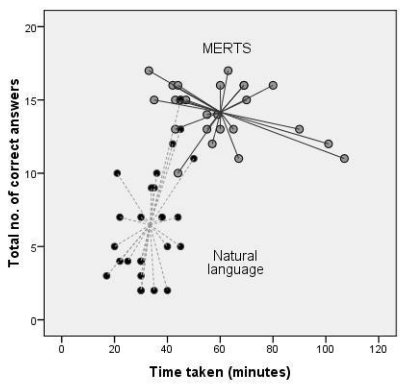

Fig. 5. Number of correct decisions versus time taken.

Fig. 5 shows a scatter plot of the points representing each of the subject's responses in two dimensions (the dependant variables), the total number of correct decisions and the total time taken (in minutes). The spikes show the distance from the centroid. A clear difference between the two treatments can be seen. MERTS is characterized by long times and greater number of correct decisions whereas NL is characterized by shorter times and fewer correct decisions.

However, an analysis of the ratios of the total number of correct decisions in relation to total time taken using MERTS strategy (ratio value = 0.2355) and NL strategy (ratio value = 0.1937) shows that the time taken to perform correct triage decisions utilizing MERTS is only 0.12 times more than the time to perform correct triage decisions utilizing NL.

Using MERTS strategy, the number of correct decisions far outweigh the number of correct decisions using NL, thus it can safely be stated that MERTS has a fairly equivalent efficiency compared to NL, even if at a first glance MERTS may seem much more resource demanding.

Nevertheless, the subjects in the experiment that used MERTS did spend more time in total, and per correct decision, even if the latter was only marginal. A potential explanation could be that the subjects using MERTS had to explicitly write a motivation/explanation referring to the strategy for every answer. This qualification of
their decisions was not present on the NL side to the same extent as the NL strategy formatting was less exact the motivations were more of the character “could be good to have”. The main motivation for demanding a thorough mapping between answers (choosing to accept or dismiss a requirement) and the MERTS formulated strategy was to enable decision traceability, a added value of MERTS that is not a part of the evaluation presented in this paper.

This might not explain the entire time difference, but at least parts of it. Using Mann-Whitney U, significance less than 0.001 was attained, indicating that there is a significant difference between the means of the two groups. This means that the null hypothesis $H_{o\ Efficiency}$ is rejected, and thus $H_{1\ Efficiency}$ is confirmed i.e. i.e. using MERTS is significantly different from NL for requirements triage with regards to efficiency. However, it cannot be concluded that use of MERTS for correct triage decisions is superior to the use of NL strategy with regards to efficiency. If the hypothesis was formulated as efficiency per correct answer, and if the time taken to write explicit qualification for the MERTS group was taken into consideration we feel confident that MERTS would be as efficient as NL, if not more.

7 Conclusions

MERTS is intended to aid product managers in performing requirements triage effectively and efficiently in a repeatable manner providing traceable decisions.

In this experiment the effectiveness and efficiency of requirements triage using MERTS was compared to using NL formulated strategies, which is the norm in industry. The main motivation of the experiment was to validate MERTS prior to industry piloting as such real industry tests require valuable and hard to obtain resources.

The experiment subjects were given 18 feature level requirements and asked to accomplish a considerable amount of work in a relatively short amount of time. The subjects were expected to form an understanding of the concept of product strategy and requirements triage, understand the domain (read and understand the requirements) and then take decisions whether to include or exclude a specific requirement based on the strategy supplied. The subjects were offered very little training, and they also possessed little prior knowledge regarding the domain compared to the level of a product manager in industry. Considering these aspects and the total number of correct decisions that resulted in using MERTS we feel that it is safe to draw the conclusion that MERTS is far superior to NL when it comes to strategy formulation and utilization for the purpose of requirements triage. The only potential drawback is that MERTS seems to be more resource intensive to use, although per correct answer we think that MERTS is at least as efficient as the NL option. Moreover, MERTS is essentially a systematic method for thinking and making decisions and that is why it takes more time but avoids errors. This systematic work is missing when using NL strategies.

The characteristics of industry are also relevant as real-life requirements triage utilizing product strategies would probably be easier for product managers than for the subjects in this controlled experiment. In industry, requirements triage and selection is not performed in isolation, regular meetings as well as official and unofficial conversations and discussions help in sharing views and reaching consensus. The benefit of MERTS is the ability to document the strategies (and the consensus) in a way that offers explicit decision support for all decision makers when performing requirements triage.

Considering these aspects, the results revealed through this experiment appear even more promising. In addition, product managers in industry are well versed in both their specific domain and in the field of requirements engineering. Given this, the use of MERTS would likely ensure even greater effectiveness and efficiency than was observed during the controlled experiment presented in this paper.

References

1. Davis, A.M.: The art of requirements triage. IEEE Computer 36 (2003) 42-49
2. Simmons, E.: Requirements Triage: What Can We Learn from a "Medical" Approach? IEEE Software 21 (2004) 86-88
3. Carmel, E., Becker, S.: A process model for packaged software development. Engineering Management, IEEE Transactions on 42 (1995) 50-61
4. El Emam, K., Madhavji, N.H.: A field study of requirements engineering practices in information systems development. Proceedings of the Second IEEE International Symposium on Requirements Engineering. IEEE Computer Society, Los Alamitos CA (1995) 68-80
5. Karlsson, L., Dahlstedt, Å., Natt och Dag, J., Regnell, B., Persson, A.: Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study. Proceedings of the Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02). Universität Duisburg-Essen, Essen, Germany (2003) 101-112
6. Gorschek, T., Wohlin, C.: Requirements Abstraction Model. Requirements Engineering journal 11 (2006) 79-101
7. Potts, C.: Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software. Proceedings of the Second IEEE International Symposium on Requirements Engineering IEEE, Los Alamitos (1995) 128-130
8. Scott, G.M.: Top priority management concerns about new product development, Vol. 13. The Academy of Management Executive (1999)
9. Krishnan, V., Karl, T.U.: Product Development Decisions: A Review of the Literature. Manage. Sci. 47 (2001) 1-21
10. Khurum, M., Aslam, K., Gorschek, T.: MERTS – A method for early requirements triage and selection utilizing product strategies. APSEC 2007, Nagoya Japan (2007)
11. Gorschek, T., Garre, P., Larsson, S., Wohlin, C.: A Model for Technology Transfer in Practice. IEEE Softw. 23 (2006) 88-95
12. Kappel, T.A.: Perspectives on roadmaps: how organizations talk about the future. Journal of Product Innovation Management 18 (2001) 39-50
13. Fenton, N.E., Pfleeger, S.L.: Software Metrics - A Rigorous & Practical Approach. International Thomson Computer Press (1996)
14. Berander, P.: Evolving Prioritization for Software Product Management. APS, Vol. PhD. Blekinge tekniska hogskola (2007)

Chapter 6 - A Method for Alignment Evaluation of Product Strategies among Stakeholders (MASS) in Software Intensive Product Development

Mahvish Khurum, Tony Gorschek
Blekinge Institute of Technology
Department of Systems and Software Engineering
S-372 25, Ronneby, Sweden
0046765500603
mkm@bth.se, tgo@bth.se

ABSTRACT

Current practices in industry are moving towards the market-driven development of software intensive products compared to customer-specific system development. Consequently, product management is faced with several challenges that have to be addressed as a part of the market-driven requirements engineering (MDRE) process. One of the important challenges is how to select the right mix of requirements, balancing short-term and long-term gains. One way to address this challenge is to utilize product strategies for selecting requirements. However, in order to do this the internal success-critical stakeholders (SCS) involved in strategies creation and requirements selection need to be aligned with respect to a product's strategic goals and objectives. This paper presents a method to enable the evaluation of degree of alignment between SCS with respect to the understanding and interpretation of a product's strategy. Further, the method not only enables the evaluation of alignment, but also specifically shows misalignment, and enables the identification of leading causes. The method has been developed in collaboration with industry and the application of it is shown through a case study at Ericsson AB.

Keywords

Product value, strategy alignment, product strategy, strategic product management, technical product management, market-driven software intensive product development

1. INTRODUCTION

With the emergence of markets for off-the-shelf/packaged and embedded software [1, 2], market-driven development of software and software intensive products is gaining increased interest/attention compared to customer-specific system development [3, 4]. Consequently, a shift in focus is occurring, affecting software development in general and requirements engineering and product management in particular [4].

In a market-driven environment the development organization takes all the risk as development is not contractually bound; rather customers are whole markets and there is a large number of *potential* customers [5]. In addition, the requirements coming in are from a wide variety of sources, both external such as market surveys and key customers, and internal sources like developers, sales, marketing, support, competitor analysis, and management [6]. This presents several challenges to the product management organization, which has to be handled as a part of the market-driven requirements engineering process (MDRE):

- First, large quantities of requirements, sometimes numbering in the thousands or even tens of thousands, risk to overload the development organization [3], thus initial triage of requirements is necessary [4, 7].

- Second, the analysis and trade-off between requirements dictates long-term vs. short-term product development, as well as the ability to balance functional requirements with non-functional aspects such as architectural longevity and maintainability.
- Third, once analyzed and weighed, the ultimate selection of what requirements to realize, and which to postpone and dismiss, are central to both short-term and long-term success of a product [8].

In this environment key-customer requirements, securing short-term revenues, are often premiered over long-term requirements, which are generally associated with higher risk. The same goes for key-customer requirements in comparison with non-functional aspects such as architectural coherence and maintainability, even though the non-functional aspects in the long run might enable savings equal or greater to the short-term revenues. The ability to balance short-term and long-term requirement selection is paramount, but time-to-market pressure, dominant in market-driven development, and pressure for quarterly revenues, often results in prioritizing key-customer short-term requirements.

In a market-driven situation, product strategies are the main tool for planning and realizing the goals and objectives of a product [4, 7, 9, 10]. Thus, from a value creation perspective it is important for product management to evaluate and select requirements that not only create value for key-customers, but also value for the product and the company by using product strategies [7, 10, 11]. This implies that product strategies need to be formulated to enable product management to perform requirements triage, trade-offs, and ultimately requirements selection [7]. Equally important is the alignment between the company's upper management, the product management and the project (realization) organization, which implies that the overall strategies need to be understood homogenously, and the same strategies need to be the basis for both the planning and the development of a product [7]. This is especially important in relation to the product management organization, as the professionals working within are, through the selection of one requirement over another, the executive arm of upper management, realizing product strategy during the market-driven requirements engineering activities. Thus, it is vital to evaluate the degree of alignment between all involved internal success-critical stakeholders (SCS) [12], which include upper management, product management, and the realizers in the project organization. A homogenous understanding as well as agreement in relation to prioritization between SCSs should be achieved to guarantee one vision through product strategies.

This paper presents a Method for Alignment Evaluation of Product Strategies among Stakeholders (MASS), which was developed in collaboration with industry, and is further illustrated through a case-study at Ericsson AB. MASS enables the evaluation of the degree of alignment between SCS with respect to the understanding and interpretation of a product's strategy. Further, it not only enables the evaluation of alignment, but also specifically shows misalignment, and enables the identification of leading causes.

The paper is structured as follows. Section 2 presents related work and a brief review of relevant literature. Section 2.2 outlines the research objectives and corresponding research questions. Section 3 presents the steps involved in MASS. The execution of MASS in an industry case study at Ericsson AB is presented in Section 4, and finally, Section 5 concludes the paper.

2. BACKGROUND AND RELATED WORK

Assessing, achieving and maintaining strategic alignment have been discussed in a number of studies [13-18] in the information systems (IS) domain. Table 1 compares MASS with other methods/frameworks by identifying the *aims and objectives* of each, the *focus area* (IS or software product), *type of strategy* used to evaluate alignment (business strategy or product strategy) and *perspective used* to evaluate alignment (projects, business and IT or software product). Looking at Table 1, it can be seen that the focus (column three) of most methods/frameworks is on the alignment of IS to the business strategies of the organization using the IS. MASS, on the other hand, focuses on the software product and the product strategy created by the development company and not the user’s (or in the market-driven perspective, the customers) business strategy. Further, looking at the perspective (column five) the focus of MASS is on the product level, not on the limited project perspective, or on any one customer’s business perspective.

A market-driven product development company has to look beyond the view of any one customer, but also beyond the internal project perspective, and focus on product and company perspectives [10]. This ensures that there is a common understanding of the company’s goals and objectives for a particular product. To the best of our knowledge no other framework has been presented to assess, achieve and maintain strategic alignment within a market-driven software product development organization.

Table 1 – Comparisons of Alignment Evaluation Methods/Frameworks

Study	Aim/Objective	Focus	Type of strategy	Perspective
[13]	Determine alignment levels by means of SA practice	IS	Business	Projects
[14]	Measure existing use of IT in organizations	IS	Business	Business and IT
[15]	Measure alignment for small firms and investigate factors that influence alignment	IS	Business	Business and IT
[16]	Identify recommendations for improving alignment based on the organization's maturity	IS	Business	Business and IT
[17]	Identify specific recommendations for improving alignment	IS	Business	Business and IT
[18]	Measuring the social dimension of alignment	IS	Business	Business and IT
MASS	Determine degree of alignment between SCS for creating product value aligned with a product’s strategy	Softwar e product	Product Strategy	Software product

2.1 Product Strategy

The issue of strategy, and in particular the different elements of a product strategy, have been visited by a number of authors. Oliver [21] broadly defined business strategy as, "*the understanding of an industry structure and dynamics, determining the organization's relative position in that industry and taking action either to change the industry's structure or the organization's position to improve organizational results*"**.** This is quite close to the definition of product strategy given by McGarth: **"***Product strategy begins with a strategic vision that states where a company wants to go, how it will get there, and why it will be successful*." [19]**.**

In order to formulate a product strategy there are a number of questions that need to be answered. The first question, "**where do we want to go?**" requires finding out the right balance between the long term opportunities [19] or goals [20] and short term objectives [19]. The basic aim of the goals is to set the general directions of movement, whereas objectives state the specific measures of accomplishment [20]. The goals refer to *profit*, *growth*, and *market share,* which potentially can be conflicting. Therefore, the product strategy normally focuses on only one of the goals at a time. As Lehmann and Winer [21] point out, if the goal/objective is to achieve a simultaneous increase in growth and profits, it is unrealistic. To attain reasonable growth requires either an increase in expenditures or decrease in profit margins [21]. Therefore, depending on the products' life cycle stages, one of the goals would have priority.

The answer to the second question, "**how will we get there?**" formulates the core of the product strategy [22]. It addresses aspects such as *customer targets, competitive targets*, and *differential advantage*. The choice of *customer targets* depends on the nature of the goals and objectives selected when answering "where an organization wants to go". For example, if the goal is to increase profits, the customer targets are the existing customer groups [21]. However, as Krishnan and Karl [23] point out, if the goal is to increase the market growth/size, the targeted customers will come from a new segment of population. Market segmentation with respect to *product's usage rates, customer/user capabilities, technology preferences, demographics*, and *purchasing power* are examples of important aspects to be considered when selecting customer targets. The choice of customer targets plays an important role in the requirements selection as the chosen customer targets set the boundaries of a product strategy, and thus sets up the rules for requirements triage, trade-off, and selection.

In order to answer "how will we get there", it is important to select *primary competitive targets,* thus prioritizing competitors [21, 24]. For determining a product's position in the market it has to be differentiated based on either cost/price or value of product offering. This means that the product has to be either low priced backed by low costs or better than competitors' products as seen by customer.

According to Lehmann and Winer [21] question three, "**what to do?**" addresses specific programs, "rules of the road," or tactics to be used to achieve goals and objectives established in the light of "how will we get there". This deals with the *product, pricing, promotion, distribution,* and service [21]. This can also be in the form of specific considerations posed by upper management. The answer to the question also decides the selection of *strategic drivers*, from amongst the *technology-push*, or *market-pull* or both.

McGarth considers question four, "**why would we be successful?"**, to be the most pertinent question to be answered to produce a competitive product strategy [19]. The answer to this question is basically related to the differential advantage aspect of the product positioning. A solid product strategy needs to provide concrete arguments for the reason of its success in the light of customers' preferences and competitive targets. For example, if the strategy is low price, this has to be proven to be an adequate differential advantage with regards to competitors.

Finally, question five, "**when will we get there?"** can be answered by roadmaps as suggested by Kappel [22]. He points out that a roadmap is a relatively common way of representing targets based on development in the context of time and releases [22].

Looking at the five questions none of them answers what important technical aspects of a software product should be considered in a product's strategy. This is handled by MERTS [7] , thus MASS also utilizes MERTS. MERTS serves two main purposes. First, it acts as a stepwise guide to creating product strategies taking both strategic and technical views into account. Secondly, the strategies resulting from MERTS can be used by managers to perform requirements triage, in essence selecting the "right" requirements for realization [7].

To summarize, the five central questions are an intricate part of the creation of a product strategy, but the understanding and the interpretation of the answers to them are at the core of product strategy alignment. Any framework for assessing, achieving, and maintaining alignment has to utilize this fact, which will become evident in the following sections as MASS is presented and validated in the case study.

2.2 Research Questions

The research questions posed as a part of this paper can be seen as both traditional research questions in light of the research being conducted and as the core of what is answered in the use of MASS at any company producing software intensive products. Table 2 gives an overview of the research questions that are used in the subsequent sections.

Table 2- Objectives of the case study and corresponding research questions

Objectives	**Research Questions (and MASS evaluation questions)**
Investigate if the SCSs understand and agree on a product's strategic goals and objectives.	RQ1: How do the SCSs understand and interpret different aspects of a product's strategy?
Determine the degree of alignment in the priority given to the goals and objectives of a product strategy between SCSs. This is assessed in three parts as follows: 1. First to understand to what degree the two groups are aligned in how they perceive the use of the strategic goals and objectives today (referred as current from now onwards). And to what degree are the SCSs within a group aligned in how they currently perceive the use of the software product's strategic goals and objectives. 2. Secondly, to understand to what degree the two groups are aligned in how they perceive the priority of the current strategic goals and objectives; and to what degree are the SCSs within a group aligned in how they currently perceive the priorities on the software product's strategic goals and objectives 3. Lastly, to what degree are the SCSs between the two groups and among the product realization group aligned with respect to ideal priorities of a product's strategic goals and objectives?	RQ2: What is the degree of alignment between and among the two groups with respect to the product strategy?
	RQ2.1: To what degree are the two groups aligned in how they perceive the use of the software product's strategic goals and objectives? RQ2.2: To what degree are the SCSs within a group aligned in how they perceive the use of the software product's strategic goals and objectives?
	RQ2.3: To what degree are the two groups aligned in how they currently perceive the priorities on the software product's strategic goals and objectives? RQ2.4: To what degree are the SCSs within a group aligned in how they currently perceive the priorities on the software product's strategic goals and objectives?
	RQ2.5: What degree are the groups aligned with respect to ideal priorities of a product's strategic goals and objectives? RQ2.6: What degree are the SCSs within a group aligned with respect to ideal priorities of a product's strategic goals and objectives?

3. MASS

Figure 1 gives an overview of MASS and its five main steps. The goal of MASS is to specify concrete steps to be followed in order to evaluate alignment between the SCSs with respect to product strategies, and in case of misalignment, identify possible causes. Each step is described in detail below.

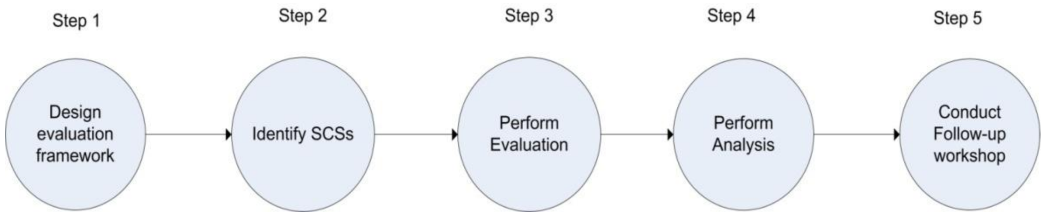

Figure 1 – MASS Steps

3.1 Step 1 – Design Evaluation Framework

The first step is to design the evaluation framework making it possible to seek answers to RQ1 and RQ2. This is demonstrated in detail in Figure 2, which shows that the evaluation framework used in MASS and that it includes both qualitative interviews and a quantitative questionnaire. RQ1 is sought through qualitative interviews utilizing the VMOST [25], MERTS and roadmapping documentation. VMOST, an organizational strategy analysis technique, is widely used to deconstruct business/product strategy and understand strategic aspects from different groups' perspectives. It is considered to be the most comprehensive technique for capturing and confirming the current strategy of a product [19]. MERTS suggests that good product strategies should contain answers to the questions stated in Section 2.2. However, the questions in VMOST mostly focus on traditional product management (business and market), whereas the technical aspects important for software product management are missing [4, 26]. Therefore, questions related to technology, architecture, and software quality need to be added as suggested in MERTS. A detailed account of the qualitative interview questions (i.e. Part 1 in Figure 2) used in MASS evaluation can be seen in Table 3, where the combination of VMOST and MERTS is evident.

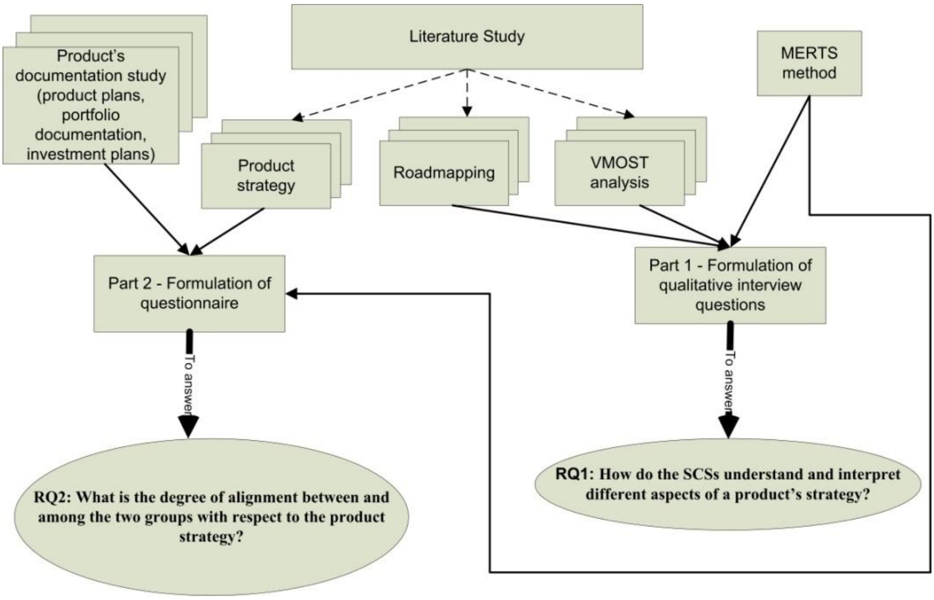

Figure 2 – Seeking answers to RQ1 and RQ2

In order to cover the second main part (i.e. Part 2 in Figure 2), and answer RQ2, documentation tied to the targeted products needs to be studied. To determine if SCSs are aligned with respect to a product's strategy it is required to identify the most important product strategy goals (in a product's strategy), which are subsequently used in the questionnaire to be prioritized by the SCSs. It is important to ensure that all the goals identified are at the same level of abstraction (i.e. strategic); otherwise it becomes difficult to compare them. For example, in a product's strategy, some goals could be at tactical level e.g., increasing strategic alliance, while others might be more abstract (strategic) e.g., increasing market share.

Table 3 – Qualitative interview questions

Method/ Technique	**Corresponding Questions**	**To Answer**
VMOST	1. What is the ideal end-state towards which the organization strives through the case product (vision)? 2. What is the primary activity that the organization performs to achieve the end-state?	Where
	3. How are the responses to questions 1 and 2 (vision and mission) appropriate and relevant to the environment? 4. What are the basic activities and their rationale by which organization competes with industry rivals?	How
	5. Are the responses to questions 1 and 2 (vision and mission) explicit or implicit? How?	Where
	6. What goals does the organization set to determine if it is competing successfully? 7. What activities does the organization perform to achieve the goals in 6? 8. How do the goals in 6 support the responses to 1? 9. What are the measurable objectives that indicate achievement of goals identified in and what activities does the organization perform to achieve those objectives? 10. How do the objectives identified in 9 support the goals identified in 6?	How
MERTS	11. Is technology innovation taken into consideration for a product strategy? 12. How is technology innovation taken into consideration for a product strategy?	How and What
	13. Is the existing architecture taken into consideration for a product strategy? 14. How is the existing architecture taken into consideration for a product strategy? For example, through, formal architectural documentation, informal discussions, and/or discussions in meetings?	What
	15. What types of roadmaps are created for a product? 16. Do you have technology roadmaps in some format?	When

In the questionnaire, the SCSs should be allowed to add more goals if they consider it necessary. Definition of each of the goals needs to be provided in the questionnaire to avoid ambiguity. In order to answer RQ2, MASS requires participants to:

a. Indicate the use of each strategic goal or objective with the values: 0 not used at all, 1: occasionally used, 2: almost always used, 3: a must (to answer RQ2.1 and RQ2.2).

b. Utilize 100-dollar method to indicate (1) how important each strategic goal or objective currently in the specific product strategy (to answer RQ2.3 and RQ2.4) and (2) how important each strategic goals or objective ideally (to answer RQ2.5 and RQ2.6).

Which of the following goals are used most when creating/realizing a product's value and what is their relative importance?

Value dimension	**Usage Today** 0 = not used, 1 = used occasionally, 2 = almost always used, 3 = is a must.	**Today**	**Points**	**Ideal**	**Points**
1. Goal 1					
2. Goal 2					
3. Goal 3					
4. Goal 4					
5. Goal 5					
		Point remaining:	**100**	**Point remaining:**	**100**

Table 4 – Questionnaire template

The 100-dollar method asks participants to spend 100 points across all of the goals given, to represent their relative influence. For example, if a participant thought a goal e.g. low production cost does not matter at all currently and achieving better quality was twice as important as increasing market share they might award these goals zero, 20 and 40 respectively. The questionnaire template is shown in Table 4.

3.2 Step 2– Identify SCS

In order to evaluate alignment, the SCS need to be identified and selected for participation. The SCS are broken down into three main groups: *strategy formulators*, *strategy realizers*, and *strategy implementers* (see Figure 3). The managers involved in the formulation of the product portfolio and individual product strategies are identified as strategy formulators. Product management, which ensures that all decisions related to development are in line with a product's strategy, and communicates these decisions to the development teams, are identified as strategy realizers. Finally, since the developments in a product should be aligned with the specific product's strategy the development teams are identified as strategy implementers.

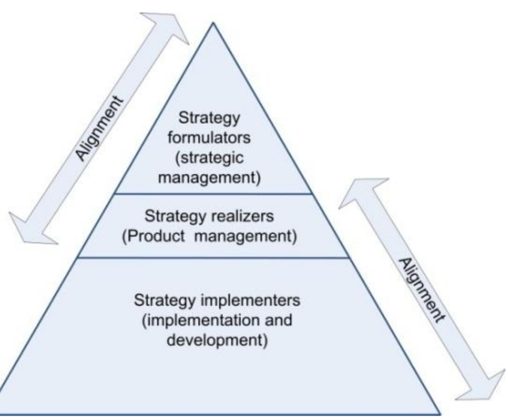

Figure 3 - Strategic Pyramid from Formulation to Implementation

MASS is focused on studying the alignment between the SCS involved in strategic management and product management (see Figure 3). The understanding and interpretation of a specific product's strategic goals and objectives are studied, and the alignment of views between strategy formulators and strategy realizers are investigated. In addition to this, internal agreement (between strategy realizers) is evaluated.

3.3 Step 3 – Perform Evaluation

After the evaluation framework is prepared and the SCSs are identified, the next step is to perform the evaluation itself. It is important to reserve at least one hour with each of the SCSs to perform both the qualitative interviews and get the questionnaire completed.

3.4 Step 4 – Perform Analysis

The purpose of this step is to analyze the data collected through the qualitative interviews and the quantitative questionnaire. The qualitative interview answers of each interviewee are coded using the Matthew/Huberman methodology [27]. The responses can be categorized as shown in Table 5, with respect to the five strategic questions and the important technical aspects previously shown in

Table 3.

With respect to each category similarity/dissimilarity is coded as “similar”, “almost similar”, and “not similar” (see the last row). In order to demonstrate this Table 5 is populated by a hypothetical example.

To what degree the groups are aligned is calculated pair-wise using a Spearman rank correlation matrix for the responses collected through the questionnaire. The correlation values will help in identifying the degree of alignment/misalignment which can be used further to elicit the possible root causes of major misalignments.

Table 5 – Categorizing qualitative interviews data

SCSs	**Vision and mission (Where)**	**Competitive strategy (How)**	**Documentation (Why)**	**Measurable goals and objectives (How and what)**	**Innovation (How and What)**	**Software quality (How and What)**	**Architecture (What)**	**Roadmapping (When)**
SCS1	To be the leader	Focus on secondary competitor	Product strategy document contains statements about vision and mission	We do not measure progress towards the goals	Innovation should be explicitly considered	Should not be considered in a product’s strategy rather it is the responsibility of development department	Is informally considered	There are product roadmaps that are followed
SCS2	To be the leader	Focus on secondary competitor	Product strategy document contains statements about vision and mission	We do not measure progress towards the goals	We do not focus on innovation thus it should not be explicitly considered	Should be stated explicitly in a product’s strategy	Not considered at all	Roadmaps are followed to a certain extent
Views	*Similar*	*Similar*	*Similar*	*Similar*	*Not similar*	*Not similar*	*Not similar*	*Almost similar*

3.5 Step 5 – Conduct Follow-up Workshop

The rationale behind potential misalignments, needed to explain and elaborate on the reasons leading to the root cause, needs to be collected post-analysis. This is the purpose of Step 5. In Step 5 the results of the interviews and the quantitative questionnaire are presented to the SCSs in a workshop setting, allowing for discussion and the collection of rationale.

One of the main reasons for the workshop is to discuss misalignments, as a first step to gain deeper understanding of the root cause(s), and to begin to homogenize interpretations, as well as change the formulations of a product's goals and strategies when relevant.

As a part of this workshop a follow-up questionnaire is used. An example follow-up questionnaire template is shown in Table 6 and Table 7. As exemplified in Table 5, first row, the "not similar" views found after categorization of the qualitative interview data are listed in the "Results" column, and corresponding questions that can be posed are shown in the second "Question" column. This can be done for all the perspectives for which the SCSs' answers are found to be "not similar".

Table 6 – Follow-up questionnaire template for qualitative interview results

Results	Questions
It has been observed in the study results that on innovation perspective, the SCSs did not agree (taken from hypothetical example shown in Table 5) Some think it should be considered explicitly and some think it should not be	Should innovation be explicitly considered or not in the product strategy? Yes No
	Why do you think it should/should not be considered in the products strategy?
	What in your view is the reason for this misalignment?

Table 7 – Follow-up questionnaire template for questionnaire results

Results	Questions
Correlation matrix showing the degree to which the groups are aligned in how they perceive the usage of strategy goals for Product A	It can be seen from the matrices on the left side that the SCSs are in most cases rather weakly aligned or in some cases negatively aligned Please tick the reasons for this misalignment from your point of view

Stakeholders	**SCS1**	**SCS2**	**SCS3**	**SCS3**
SCS1		XX	YY	ZZ
SCS2			XY	YZ
SCS3				XZ
SCS3				

Correlation matrix showing the degree to which the groups are aligned in how they perceive the priority of strategy goals for Product A

Stakeholders	SCS1	SCS2	SCS3	SCS4
SCS1		XX	YY	ZZ
SCS2			XY	YZ
SCS3				XZ
SCS4				

Correlation matrix showing the degree to which the groups are aligned in how they perceive the priority of strategy goals for Product A

Stakeholders	SCS1	SCS2	SCS3	SCS4
SCS1		XX	YY	ZZ
SCS2			XY	YZ
SCS3				XZ
SCS4				

1. Product size and lack of communication
2. Lack of measurable objectives translating strategy into action
3. Lack of feedback cycle from product/project experiences to update strategy
4. Product strategy does not explicitly explain the strategic factors and their prioritization
5. Lack of a specific perspective in a strategy (e.g. lacking technology perspective)
6. Lack of understanding of a company's strategy
7. Any other….

Results	Questions
Overall prioritization of the five strategic factors according to their relative importance currently (from most important to least important). 1. Factor A 2. Factor C 3. Factor B 4. Factor E 5. Factor D	Do you agree A is the most important factor? Yes No
	Why?
	Do you agree D is the least important factor? Yes No
	Why?
	Do you agree B should be most important factor? Yes No
Overall prioritization of the five strategic factors according to their relative importance that should be ideally (from most important to least important) 1. Factor B 2. Factor C 3. Factor D 4. Factor A 5. Factor E	Why?
	Do you agree E should be the least important factor? Yes No
	Why?

Table 7 shows the correlation results of the quantitative data (from the quantitative questionnaire) and the corresponding questions that can be posed in the workshop follow-up questionnaire to investigate the root cause for any misalignments.

3.5.1 Misalignment Analysis

There can be several interdependent reasons for misalignment. To aid participants in Step 5, the following reasons can be presented (as shown in Table 7) to initiate a discussion. These reasons can be [13-18, 28]:

1. Product size and communication quality: Effective exchange of ideas and a clear understanding of what it takes to ensure successful strategies are high on the list of enablers and inhibitors to alignment [28]. Since for larger products more people are involved to manage the product, which could be a reason for further impairment of communication quality.
2. Lack of measurable objectives: Another reason for misalignment could be the lack of consensus as to how a strategy should be implemented, and how to measure that the strategy goals are being met. Absence of measurable objectives at the product management was confirmed by the participants. It is understandable that business and technology metrics differ because they are different in nature [28], however it is important to translate business strategy not only into business measurable objectives (i.e., financial) but also into technical measurable objectives.
3. Missing feedback cycle: It is not enough to have measurable objectives, it is equally important to use the metrics and collected data to provide feedback on the strategy and make changes accordingly. The misalignment between the ideal priorities of strategic goals could potentially be due to the fact that there is no mechanism to adjust strategies in future based on previous experiences.

4. Lack of technical perspective: One of the reasons for misalignment between SCSs with respect to the technical goals could be the fact that technical aspects are not brought to the table when product strategies are formulated. As a result, goals related to the technical aspects e.g. software quality, innovation, architectural considerations and technology roadmapping are not explicitly represented and prioritized.
5. Lack of understanding of a company's strategy: If a company's vision and mission statements in the company's strategy are not understood at each strategic level (strategy formulation level and strategy realization level), chances are great that product strategies do not reflects company's direction of movement which can further misalign strategy formulators and strategy realizers with respect to a product's specific goals and objectives.

   Participants are allowed to add any additional factors that are not covered in the list. Depending on how extensive the follow-up is, the prioritized goals with respect to relative current importance can be arranged in descending order as shown in last two rows of Table 7.

. Corresponding follow-up questions can be posed if the participants agree or disagree with this prioritization and motivations are caught. Similarly, this can be done for the prioritized goals with respect to their relative importance in a “wish” or “ideal” case proposed by the participants.

The difference and distance between the judged current priorities (“what priority do you perceive today”), and the ideal priorities (“how do you think it should be”) can also be investigated. Analysis of the priorities and the stated rationale can help to identify the root cause for misalignment, as well as jump start activities for homogenization.

After the follow-up the data collected through the questionnaire can be used to identify the common causes of misalignments and the dependency between them to further identify the root causes. This has been demonstrated in the case study (see Section 4.1.5).

4. MASS AT ERICSSON – A CASE STUDY

The case study was conducted during the autumn of 2008 involving three Ericsson products (designated Product X, Y, and Z due to reasons of confidentiality). Ericsson is a world leader in the telecommunications sector, providing a wide range of products and solutions. Products are developed and sold as generic solutions offered to an open market, although customized versions of the products are also developed for key customers. Case products’ characteristics are given in Table 8. It is important to note that the findings through MASS application are specific to the case products and thus cannot be generalized for all Ericsson's products.

Table 8 – Case Products' Characteristics

Products	**Size**	**Maturity (in terms of years)**	**Total releases**	**Release frequency**
X	Large	9	7	1 per year
Y	Small	4	1	1 per 4 years
Z	Medium	9	4	1 per 2 years

4.1 Execution

The subsections below mirror the description of MASS step-by-step as given in Section 3.

4.1.1 MASS: Step 1 – Design Evaluation Framework

An evaluation framework was designed using the method proposed in Section 4.1. The qualitative questions stated in Table 3 were used. Five strategic goals for the quantitative questionnaire were identified using product strategy literature and Ericsson's strategy documentation for the three case products X, Y and Z.

4.1.2 MASS : Step 2 – Identification of SCS

Members of the strategic product management organization at Ericsson supported in the identification of the roles for the two internal groups for this case study: strategy formulators and strategy realizers. For each product there was a Upper Product Manager (UPM) who is responsible for the overall strategies i.e. strategy formulation. In addition there were 1-4 Strategic Product

Managers (SPM), dealing with product release, product market and general functionality. A total of ten potential participants were identified, three UPMs, one for each product, and seven SPMs (three SPMs for Product X and two SPMs each for Product Y and Product Z).

4.1.3 MASS : Step 3 – Perform Evaluation

A round of semi-structured interviews was conducted with three UPMs and three SPMs. It was not possible to interview all the SPMs for the three products due to their busy schedules. Since the idea behind the qualitative interviews is to investigate how strategy formulators and strategy realizers understand and interpret a product's strategy, it was considered appropriate to interview at least one strategy formulator and one strategy realizer for each product, this was achieved. The interviews were initiated with the qualitative questions (see Table 3), and subsequently the questionnaire to prioritize the goals and objectives was provided (see Table 4). All of the UPMs and SPMs (even the ones not being interviewed) completed the questionnaire with the prioritizations.

4.1.4 MASS : Step 4 – Perform Analysis

4.1.4.1 Answering RQ1

The results of the categorization and similarity analysis are shown in Table 9. Each category of responses is discussed to evaluate alignment in relation to the defined categories. The specific strategic details are not reported in the paper due to reasons of confidentiality; rather the similarities and differences are discussed in a general manner.

Table 9 - Categorizing qualitative interviews data

Products	Vision and mission (Where)	Competitive strategy (How)	Documentation (Why)	Measurable goals and objectives (How and what)	Innovation (How and What)	Software quality (How and What)	Architecture (What)	Roadmap-ping (When)
X	Similar views	Similar view	Similar view	Similar view	Not similar view	Not similar view	Similar view	Similar view
Y	Similar view	Similar view	Similar view	Similar view	Almost Similar view	Not similar view	Similar view	Similar view
Z	Similar view	Similar view	Similar view	Similar view	Almost Similar view	Not similar view	Not similar view	Similar view

Looking at column two, three and four in Table 9 agreement between the UPMs and SPMs can be observed. This is true for each product with respect to the product's vision and mission ("Where"), competitive strategy ("How") and how the strategy for each product is documented ("Why"). They also agreed on the measurable goals and objectives ("How") aspect, stating that they lack measurable objectives explicitly linked to the product's strategic goals. This is discussed later in Section 4.1.5 where possible root causes for misalignments are elaborated upon.

Considering the architectural aspect in column eight ("What"), they all agreed that they do take the existing architecture of the product into account; however, they do it informally through meetings. However, they stated that they have realized the need for more explicit architecture documentation that can serve as an input to the formulation of the product strategy, avoiding surprises in the longer run.

From the innovation perspective in column six (“What”), it can be seen that the UPMs and SPMs were not completely aligned. The opinions differed on whether the innovation aspect was explicitly considered in a product’s strategy. While the UPMs were of the opinion that technology innovation is explicitly considered, the SPMs were either not sure that it was explicitly considered, or stated that it is not considered. In response to the question if technology innovation is taken into consideration for a product strategy, the UPMs mentioned that for their respective products they are relatively slow in adopting new technologies, and one of the UPMs explained: “*Innovations are mostly driven by analysts’ reports, which discuss the emerging technologies and trends with respect to their potential market share*”.

The SPMs were of the view that sources of innovation are: the analysts’ reports, Ericsson’s internal research projects and ideas from the development team. However, one of the SPMs mentioned: “A*n important criteria for success is to gain market share and if this can be done without being innovative then there is no need for innovation*”. Moreover, SPMs also agreed that innovative features are not considered as high value, at least not to the degree of being prioritized over present customer's functional requirements.

Regarding the software quality goals in a product's strategy ("What"), the UPMs and SPMs had divergent views as is evident from Table 9. The UPMs for Product X and Product Y stated: "*the quality aspect is considered only when there are quality issues, otherwise basic qualities are assumed to be taken care of as a matter of practice"*. The UPM of Product Z mentioned: "*we had a number of quality requirements stated in the product's strategy*". While the SPMs agreed that quality is not considered explicitly within a product's strategy, they mentioned/agreed that it should be explicitly stated in the product's strategy. This, they stated, can increase the perceived value and priority of quality requirements, which are usually not considered as high value in comparison with functional requirements stemming from customers.

With respect to roadmapping (last column of Table 9), i.e., the "When" part of the strategy, the UPMs and SPMs were aligned. They stated that they had product roadmap documents that were quite detailed, but they did not know of explicit technology roadmaps. However, they acknowledged the importance of it.

Summarizing the qualitative data analysis, it can be concluded that the UPMs and SPMs have a common understanding and interpretation with respect to the

"Where", "How" and "When" parts of the strategy, but their understanding differs with respect to the "What" part which involves the technical considerations of a product.

4.1.4.2 Answering RQ2

Since the product strategies are specific to products, the correlation analysis of the degree of alignment between the usage and priorities of strategy factors was carried out for each product separately.

Use of Product's Strategic Goals and Objectives. With respect to the use of goals for Product X, it can be seen from the first row of Table 10 that the degree of alignment between UPMs and the SPMs is less than 70% in all of the cases. None of the correlations are significant, therefore, it can be concluded that the degree of alignment is relatively weak. However, it is better in the cases of the UPM and SPM1 (66.7%) and the UPM and SPM3 (64.5%). The degree of alignment amongst the SPMs who are realizers of the strategy was very low.

Table 10 shows a similar analysis for Product Y. It shows that the UPM and SPM groups are quite aligned (between the UPM and SPM1= 81.1% and correlation between UPM1 and SPM2 = 89.5%). For product Z, the correlation between UPM and one SPM1 is strikingly perfect (100%). Similarly, the degree of alignment between SPMs is strong (76.1%).

Based on the correlations, it can be concluded that degree of alignment between UPMs and SPMs in how they perceive the usage of strategic goals for the three products (X, Y and Z) is strong (between 66% and 100%), though not very strong in every case. The degree of alignment among the SPMs (the realizers of strategy) varies between 0% to 97% which shows that in some cases it is very strong (Product Y), while for others it is very weak (Product X).

Table 10 – Correlation matrix showing the degree of alignment between the groups and among SPMs in relation to how they perceive the use of strategy goals currently for Product X, Y and Z

Products	Stakeholders	UPM	SPM1	SPM2	SPM3
Product X	UPM		0.67	0.15	0.64
	SPM1			0.30	0.64
	SPM2				0
	SPM3				
Product Y	UPM		0.81	0.89	
	SPM1			0.97	
	SPM2				
Product Z	UPM		1.00	0.76	
	SPM1			0.76	
	SPM2				

Priority of Product's Strategic Goals and Objectives Currently. Table 11 shows that the degree of alignment between the UPMs and SPMs, as to how they perceive the priority of strategic goals and objectives currently for the three products (X, Y and Z), is not so strong (between 63% and 74%). The UPMs and SPMs seem to agree on the use of the strategy goals, but not on their relative importance.

Table 11 - Correlation matrix showing the degree to which the groups are aligned in how they perceive the priority of strategy goals currently for Product X, Y, Z

Products	Stakeholders	UPM	SPM1	SPM2	SPM3
Product X	UPM		0.63	0.13	0.70
	SPM1			0.41	0.56
	SPM2				0.76
	SPM3				
Product Y	UPM		0.74	0.60	
	SPM1			0.95	
	SPM2				
Product Z	UPM		0.63	0.81	
	SPM1			0.58	
	SPM2				

The degree of alignment among the SPMs (the realizers of strategy) varies between 13% to 95%, which shows that in some cases it is very strong (Product Y), while for others it is weaker (Product X).

Ideal Priority of the Product's Strategic Goals and Objectives. Finally, Table 12 shows the degree of alignment between the UPMs and SPMs, in relation to their perception of the ideal priority of strategic goals for the three products. Here the agreement is rather weak (between -90% and 78%). For Product X, the correlation between UPM and SPM1 is negative, implying that they are negatively aligned in their perspectives as what should ideally be the priority of strategic goals. Among the SPMs, the degree of alignment varies between -100% to 95% which shows that in some cases it is very strong (e.g. Product Y) while for others it is negative (Products X and Z).

Table 12 - Correlation matrix showing the degree to which the groups are aligned in how they perceive the ideal priority of strategy goals for Product X, Y, Z

Products	Stakeholders	UPM	SPM1	SPM2	SPM3
Product X	UPM		-0.90	-0.10	1.00
	SPM1			0.36	0.30
	SPM2				0.56
	SPM3				
Product Y	UPM		0.78	0.74	
	SPM1			0.95	
	SPM2				
Product Z	UPM		0.15	-0.60	
	SPM1			0.05	
	SPM2				

Based on the qualitative and quantitative data analysis it is concluded that when it comes to slogans of the strategy and general statements, the UPM and SPMs generally agree. However, when the strategy is broken down to concrete goals and they are the SCSs are asked to indicate use and priorities of strategic factors (both current and ideal) the misalignment is evident and in some cases severe.

4.1.5 MASS: Step 5 – Conduct Follow-up Workshop

As suggested in Step 5 of MASS, to conduct the workshop a follow-up questionnaire was prepared based on the example follow-up questionnaire template (see Table 6 and Table 7).

Two UPMs and three SPMs were presented with the results for their respective products and they were asked to identify and discuss potential root causes for the misalignments. Due to confidentiality reasons, the exact questionnaire containing the results cannot be presented. However, common root causes identified during these feedback sessions are given below. The first and fifth root causes were not given in the original follow-up questionnaire but were added by the participants during the follow-up workshop as additional important causes for misalignment. Using the feedback from participants potential dependencies between the root causes are also discussed below.

C-1. Lack of incentives: According to two SPMs: "n*o incentives for being proactive and innovative are given*". Currently, Ericsson's approach is reactive and not proactive: for an approach to be proactive more time and effort is required, however, no time and effort is budgeted for proactive and innovative planning.

C-2. Lack of relevant communication: According to an SPM: "there is a *lot of daily discussion around the events happening on daily or weekly basis but no discussions about long term strategy between SPMs*". All the participants agreed to this reason.

C-3. Lack of explicit explanation of strategic factors and their importance in documented product's strategy: As the SPMs get statements from the UPMs, every SPM understands and acts according to his/her own interpretation of the statements.

C-4. Incorrect use of product/project experiences to update strategy: Mostly working product strategies are written down as an account of history, and not as something to be used as a planning tool. This is primarily due to the SPMs point of view that there is no incentive to work on strategies (C-1 leads to C-4). Moreover, in the absence of explicit description of strategic factors and their relative importance updating a strategy after learning from experiences does not make any sense (C-3 leads to C-4).

C-5. Event driven decisions: There are two types of strategies: documented product strategy (explicit) and working product strategy (implicit). The UPMs give equal importance to all the strategic factors in the documented product strategies, whereas, depending on the current events in the market such as competition and customers requirements, each SPM (the strategy realizer) reacts to the situation and chooses which strategic factors are important at a certain point of time (C-2 and C-3 lead to C-5)

C-6. Lack of measurable objectives: Everybody agreed that they do not have any measurable objectives that translate strategy into action. Rather there are key progress indicators that determine personal progress and salary. It can be deduced that in the absence of explicit explanation of strategic factors and their importance in documented strategies related to products, it is almost impossible to state measurable objectives (C-3 leads to C-6).

Figure 4 shows the dependency diagram of the identified causes. From the diagram, it can be deduced that C-1, C-2 and C-3 are the root causes that need to be dealt with in order to solve the issue of misalignment.

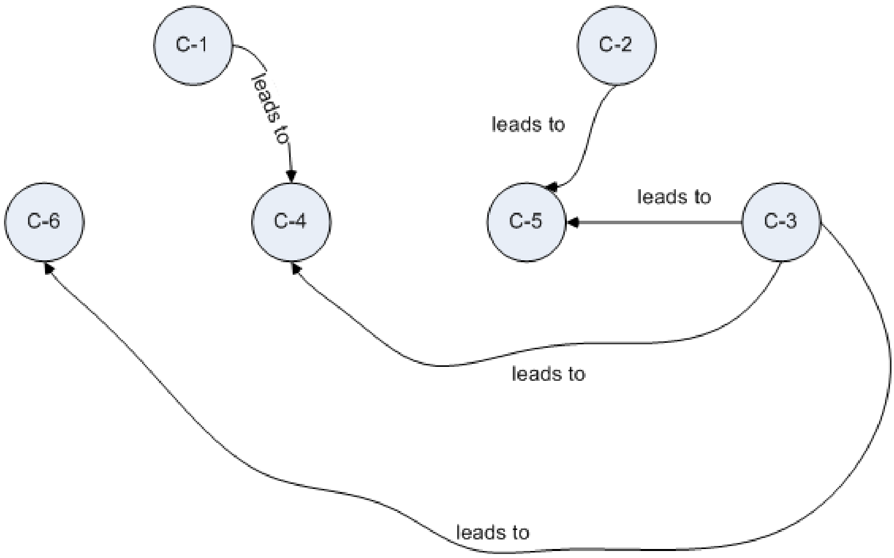

Figure 4 – Dependency Diagram

4.2 Validity Threats

In this section, we discuss the threats to the validation of MASS. We base this on the discussion of validity and threats as presented in Wohlin et al. [29]. The validity threats considered are conclusion, internal and external validity threats respectively.

4.2.1 Conclusion Validity

The sampling techniques used for identifying the SCSs can pose a threat to the validity of the evaluation. The subjects selected may not be totally representative for the role they should represent at Ericsson for the three case products. The main assurance that this misrepresentation is minimal is the fact that the subjects were selected in cooperation with two senior managers with extensive knowledge and experience concerning the development processes and the personnel at Ericsson.

4.2.2 Internal Validity

As the evaluation and feedback sessions of MASS were performed with the different interview subjects, they were called upon to voice their opinions and views regarding e.g. product strategies formulation in general and results in particular. As their answers were registered by the researcher this could have constrained people in their answers. This potential problem was alleviated by the guarantee of anonymity as to all information divulged during the evaluation and feedback sessions, and that recorded answers was only to be used by the researcher, i.e. not showed or used by any other party.

4.2.3 External Validity

The external validity is concerned with the ability to generalize the results, i.e. in this case the applicability of MASS in industry outside the case company.

MASS can be applied in any company as it is not a one-size fits all method. It details the steps to be performed for evaluation of alignment. The evaluation framework used in MASS can be tailored based on different product strategies and similarly SCSs can be identified based on who performs the roles specified by MASS.

4.3 Case Study Conclusions

Through the application of MASS at Ericsson, the qualitative interviews data analysis shows that the two SCS groups related to strategy formulation and strategy realization are aligned with respect to vision and mission statements. However, they have divergent views when it comes to the technical product aspects (innovation and quality requirements). Some think that technical aspects should be explicitly considered in a product strategy while some are of the opinion that they should not be. The analysis of the quantitative data further revealed:

1. The degree of alignment between the UPMs and SPMs in how they perceived the usage of strategic goals for the three products (X, Y and Z) was strong (between 66% and 100%), though not very strong in every case. The degree of alignment among the SPMs (the realizers of strategy) varied between 0% to 97% which shows that in some cases it was very strong (Product Y), while for others it was very weak (Product X).

2. The degree of alignment between the UPMs and SPMs as to how they perceive the priority of strategic goals and objectives for the three products (X, Y and Z) was not so strong (between 63% and 74%). The degree of alignment among the SPMs (the realizers of strategy) varied between 13% to 95% which shows that in some cases it was very strong (Product Y), while for others it was very weak (Product X).
3. The degree of alignment between the UPMs and SPMs in how they perceived the ideal priority of strategic goals for the three products (X, Y and Z) was rather weak (between -90% and 78%). Among the SPMs, the degree of alignment varied between -100% - 95% which shows that in some cases it was very strong (e.g. Product Y) while for others it was negative (Products X and Z).

When these results were presented to the participants according to Step 5 of MASS, following possible root causes of misalignment were identified:

C-1. Lack of incentives.

C-2. Lack of relevant communication.

C-3. Lack of explicit explanation of strategic factors and their importance in documented product's strategy.

The MASS evaluation has helped Ericsson identify misalignments as well as the root causes. While for large companies like Ericsson these misalignments are bound to exist (due to large products), mature companies like Ericsson can use MASS evaluations as a continuous effort for improving alignment, homogenizing understanding and agreeing on future prioritization of factors.

5. CONCLUSIONS AND FUTURE RESEARCH

In market-driven software development, product management is faced with several challenges. Requirements overload, selection of a right mix of requirements for balancing a product's short and long terms goals, and time-to-market pressure, to name a few, all are paramount for product success. Utilizing product strategies to perform requirements triage, trade-offs, and ultimately requirements selection has proven successful [7, 10, 11]. However, to ensure that that the overall strategies are understood and prioritized homogenously, and the same strategies are used as the basis for both the planning and the development of a product, alignment between the SCS needs to be assured. MASS was developed in collaboration with industry, which is illustrated through a case-study at Ericsson AB, to enables the evaluation of degree of alignment between upper management (strategy creators) and the product management (strategy realizers) with respect to the understanding and interpretation of a product's strategy. MASS shows misalignment, and enables the identification of leading causes.

MASS contains five steps that build on established technologies such as VMOST and MERTS, which were combined to cover several perspectives including the management, marketing and technical views. In addition, MASS is the first method of its kind covering the strategy evaluation and refinement for development organizations in a market-driven product development context. It focuses on the software product and the product strategy of a market-driven software development company, unlike the previous studies [13-18] which were limited to the project perspective for achievement, assessment and maintenance of strategic alignment, and in addition focused on the business strategies of the customer organization.

During the development of MASS we did not adopt a one-size-fit-all philosophy with regards to strategies, but rather by choosing the specific product's strategic goals and objectives, a tailoring towards a product (or organization) can be achieved prior to use in order to adapt to organizational and product specific goals and objectives. This makes it possible for any organization to use MASS to the degree needed, making it suitable for larger companies as well as small and medium sized enterprises.

At present MASS has been developed in collaboration, and used in one company through a case study where three product organizations were evaluated. Future work involves application of MASS in other organizations in industry. During the execution of MASS in the case company, however, we realized that the goals stated in the questionnaire, for assigning point to, were on an abstract level and as a refinement of MASS, the questionnaire could be divided into a hierarchy of strategic, tactical and operation goals. This can help the SCS to understand the goals even more clearly. In addition, future research involves looking into possible solutions for the identified root causes of misalignments. Moreover, currently MASS only focuses on evaluation of alignment and the identification of root causes for misalignment, steps needed to maintain alignment still needs to be incorporated in MASS.

ACKNOWLEDGMENTS

We would like to thank reviewers from Ericsson, Sebastian Barney and Nina D. Fogelström for reviewing the paper and providing valuable feedback which has helped in approving the paper.

REFERENCES

[1] E. Carmel and S. Becker, "A process model for packaged software development," *Engineering Management, IEEE Transactions on,* vol. 42, pp. 50-61, 1995.

[2] K. El Emam and N. H. Madhavji, "A field study of requirements engineering practices in information systems development," in *Proceedings of the Second IEEE International Symposium on Requirements Engineering*, Los Alamitos CA, 1995, pp. 68-80.

[3] L. Karlsson, Å. Dahlstedt, J. Natt och Dag, B. Regnell, and A. Persson, "Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study," in *Proceedings of the Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02),* Essen, Germany, 2003, pp. 101-112.

[4] T. Gorschek and C. Wohlin, "Requirements Abstraction Model," *Requirements Engineering journal,* vol. 11, pp. 79-101, 2006.

[5] C. Wohlin, A. Aurum, and (editors), *Engineering and managing software requirements*, 1st ed. New York, NY: Springer, 2005.

[6] C. Potts, "Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software," in *Proceedings of the Second IEEE International Symposium on Requirements Engineering* Los Alamitos, 1995, pp. 128-130.

[7] M. Khurum, K. Aslam, and T. Gorschek, "MERTS – A method for early requirements triage and selection utilizing product strategies," in *APSEC 2007* Nagoya Japan, 2007.

[8] I. van de Weerd, S. Brinkkemper, R. Nieuwenhuis, J. Versendaal, and L. Bijlsma, "Towards a reference framework for software product management," Los Alamitos, CA, USA, 2006, pp. 312-15.
[9] A. Aurum and C. Wohlin, "Aligning Requirements with Business Objectives: A Framework for Requirements Engineering Decisions," in *Proceedings of Requirements Engineering Decision Support Workshop* Paris, France, 2005.
[10] T. Gorschek and A. M. Davis, "Requirements engineering: In search of the dependent variables," *Information and Software Technology,* vol. 50, pp. 67-75, 2008.
[11] C. Wohlin and A. Aurum, "Aligning Requirements with Business Objectives: A Framework for Requirements Engineering Decisions," in *Proceedings of Requirements Engineering Decision Support Workshop* Paris, France, 2005.
[12] B. Boehm, "Theory-W software project management: principles and examples," *IEEE Transactions on Software Engineering,* vol. 15, pp. 902-916, 1989.
[13] D. Avison, J. Jones, P. Powell, and D. Wilson, "Using and validating the strategic alignment model," *Journal of Strategic Information Systems,* vol. 13, pp. 223-46, 2004.
[14] Y. E. Chan, S. L. Huff, and D. G. Copeland, "Assessing realized information systems strategy," *Journal of Strategic Information Systems,* vol. 6, pp. 273-98, 1997.
[15] A. J. Hale and P. B. Cragg, "Measuring strategic alignment in small firms," Los Alamitos, CA, USA, 1996, pp. 127-35.
[16] J. Luftman, "Assessing IT/business alignment," *Information Systems Management,* vol. 20, pp. 9-15, 2003.
[17] R. Papp and P. Coleman, ""Strategic alignment: analysis of perspectives," in *Southern association of Information Systems Conference*, Florida, 2006.
[18] B. H. Reich and I. Benbasat, "Factors that influence the social dimension of alignment between business and information technology objectives," *MIS Quarterly,* vol. 24, pp. 81-4, 2000.
[19] M. E. McGrath, *Product Strategy for High Technology Companies*, 2nd ed.: McGraw-Hill Professional Book Group, 2000.
[20] C. M. Crawford and C. A. D. Benedetto, *New Products Management*, 7th ed.: McGraw-Hill/Irwin, 2002.
[21] D. R. Lehmann and R. S. Winer, *Product management*, 3. ed. Boston MA: McGraw-Hill, 2002.
[22] T. A. Kappel, "Perspectives on roadmaps: how organizations talk about the future," *Journal of Product Innovation Management,* vol. 18, pp. 39-50, 2001.
[23] V. Krishnan and T. U. Karl, "Product Development Decisions: A Review of the Literature," *Manage. Sci.,* vol. 47, pp. 1-21, 2001.
[24] J. M. Utterback, *Mastering the Dynamics of Innovation: How Companies Can Seize Opportunities in the Face of Technological Change*: Harvard Business Press, 1994.
[25] R. Sondhi, *Total Strategy*: Airworthy Publications International Ltd, 1999.
[26] T. Gorschek, M. Svahnberg, A. Borg, A. Loconsole, J. Borstler, K. Sandahl, and M. Eriksson, "A controlled empirical evaluation of a requirements abstraction model," *Information and Software Technology,* vol. 49, pp. 790-805, 2007.
[27] M. Matthew B and A. M. Huberman, *Qualitative Data Analysis*: SAGE Publications, 1994.
[28] J. LUFTMAN, "Assessing business-IT alignment maturity," *Communications of the Association for Information Systems,* vol. 4, 2000.
[29] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*. Boston MA: Kluwer Academic, 2000.

ABSTRACT

At the core of choosing what features and level of quality to realize, and thus offer a market or customer, rests on the ability to take decisions. Decision-making is complicated by heterogeneous understanding of issues such as priority, implication of realization, and interpretations of strategy as pertaining to the short- and long-term development of software intensive systems. The complexity is further compounded by the amount of decision support material that has to be taken into account and the sheer volume of possible alternatives that have to be triaged and prioritized, thousands, or even tens of thousands of requirements can be the reality facing industry. There is a need to develop the functionality that is strategically most important while satisfying customers and being competitive, cost efficient, and minimizing risk. In order to achieve a balance between these factors, it is important that within an organization, all stakeholders agree on the strategic aspects to be considered and their corresponding relative importance.

The objective of this thesis is to provide strategic decision-making support for software product management.

A number of empirical studies, set in both academia and industry have been performed. The research methods used span from systematic reviews, case studies to experiments, all aligned to identify possibilities for improvement, devise solutions, and validate these solutions in several steps.

The methods presented in this thesis can be used to evaluate strategic alignment and identify possible root causes for misalignments. To strengthen strategic alignment, the methods and results in the thesis can be used by product managers for making effective and efficient strategic decisions regarding portfolios and products, following a systematic and aligned process.

The area of software product management in the context of market-driven software intensive product development is a field with unique challenges. The specifics of the solutions are based on industry case studies performed to gauge state of the art as well as map the main challenges. The decision-making support developed takes the form of models and methods that support software product management in strategic alignment, requirements triage and portfolio level decisions.

ISSN 1650-2140
ISBN 978-91-7295-168-6
2009:08