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 the diverse understanding of issues such as priority, consequence of realization, and interpretations of strategy as pertaining to the short-term and long-term development of software intensive products. 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 a company. There is a need to develop the functionality that is strategically most significant, while satisfying customers and being competitive, time efficient, cost effective, and risk minimizing. In order to achieve a balance between these factors, all the stakeholders, within an organization, need to agree on the strategic aspects and value considerations to be considered, and their corresponding relative importance.

Objective: The objective of this thesis is to provide enhanced decision support for product managers faced with decision-making challenges. This involves, but is not limited to, enhancing the alignment between the product and portfolio management with respect to product strategies, and enabling the use of value as a basis for product management and development related decisions.

Method: A number of empirical studies, set in industry, have been performed. The research methods used span from systematic mapping, and systematic reviews to case studies, all aligned to identify possibilities for improvement, devise solutions, and incrementally evaluate said solutions. Close collaboration with industry partners was at the core of the research presented in this thesis.

Result: The MASS method presented in this thesis can be used to evaluate strategic alignment and identify possible root causes for misalignment. To strengthen strategic alignment, the Software Value Map and corresponding decision support material, proposed in the thesis, can be used by product managers for making effective and efficient strategic decisions in relation to portfolios, products and process improvement, following a systematic and aligned process.

Conclusions: 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 identify the main challenges. The decision support developed takes the form of maps and frameworks that support software product management on product and portfolio level decisions, strategic alignment, value-based requirements selection, and value-based process improvement.
Decision Support for Product Management of Software Intensive Products

Mahvish Khurum
Decision Support for Product Management of Software Intensive Products

Mahvish Khurum

School of Computing
Blekinge Institute of Technology
SWEDEN
To my Allah, for blessing me with the abilities and opportunities;
To my family, for their continuous support, love and prayers.
First, I would, sincerely, like to thank my supervisors, Dr. Tony Gorschek and Professor Claes Wohlin, for their valuable feedback, expertise and advice. They have helped me grow as a researcher and a person. I especially thank them 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: Dr. Kai Petersen, Dr. Sebastian Barney, Magnus Wilson and 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. Finally, I would like to thank all my family members who supported me. Special thanks to my husband, Khurum Aslam, who made it possible for me to pursue my dreams.

The research is funded by The Knowledge Foundation in Sweden under a research grant for the project Blekinge Engineering Software Qualities (BESQ) and Ericsson Software Research.
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 the diverse understanding of issues such as priority, consequence of realization, and interpretations of strategy as pertaining to the short-term and long-term development of software intensive products. 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 a company. There is a need to develop the functionality that is strategically most significant, while satisfying customers and being competitive, time efficient, cost effective, and risk minimizing. In order to achieve a balance between these factors, all the stakeholders, within an organization, need to agree on the strategic aspects and value considerations to be considered, and their corresponding relative importance.

Objective: The objective of this thesis is to provide enhanced decision support for product managers faced with decision-making challenges. This involves, but is not limited to, enhancing the alignment between the product and portfolio management with respect to product strategies, and enabling the use of value as a basis for product management and development related decisions.

Method: A number of empirical studies, set in industry, have been performed. The research methods used span from systematic mapping, and systematic reviews to case studies, all aligned to identify possibilities for improvement, devise solutions, and incrementally evaluate said solutions. Close collaboration with industry partners was at the core of the research presented in this thesis.

Result: The MASS method presented in this thesis can be used to evaluate strategic alignment and identify possible root causes for misalignment. To strengthen strategic alignment, the Software Value Map and corresponding decision support material, proposed in the thesis, can be used by product managers for making effective and efficient strategic decisions in relation to portfolios, products and process improvement, following a systematic and aligned process.

Conclusions: 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 identify the main challenges. The decision support developed takes the form of maps and frameworks that support software product management on product and portfolio level decisions, strategic alignment, value-based requirements selection, and value-based process improvement.

Keywords
Software product management, Value-based decision support, consolidated view of value, decision-making, strategic alignment, requirements selection, software process improvement, technology product management, software intensive systems.
Overview of Papers

Papers included in the thesis


Papers not included in the thesis


9. **Requirements management for continuous software product development**
   Mahvish Khurum, Sebastian Barney, Nina D Fogelström, Tony Gorschek,

10. **Supporting Students Improve with Rubric-Based Self-Assessment and Oral Feedback**
    Sebastian Barney, Mahvish Khurum, Kai Petersen, Michael Unterkalmsteiner, Ronald Jabangwe,
    Accepted for publication in: IEEE Transactions in Education, 2011.

11. **The Mythical Adequate Level of Details - Bundling Requirements for Large Scale Market Driven Engineering**
    Daniel Lucas Hirtz, Mahvish Khurum,
## Contents

### Chapter 1 - Introduction

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Positioning the Research</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Definitions</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Software Product Management</td>
<td>3</td>
</tr>
<tr>
<td>1.3 An Overview of Decision-making</td>
<td>6</td>
</tr>
<tr>
<td>1.4 Portfolio Management</td>
<td>11</td>
</tr>
<tr>
<td>1.5 An Overview of Product/Market-driven Requirements Engineering</td>
<td>13</td>
</tr>
<tr>
<td>1.6. Processes and Quality</td>
<td>18</td>
</tr>
</tbody>
</table>

### 2. Related Work

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 C1: Identification of Appropriate Domain Analysis Solutions</td>
<td>22</td>
</tr>
<tr>
<td>2.2 C2: Creation of Product Value</td>
<td>22</td>
</tr>
<tr>
<td>2.3 C3: Alignment with respect to Product Strategy</td>
<td>23</td>
</tr>
<tr>
<td>2.4 C4: Selection of Innovative Features</td>
<td>23</td>
</tr>
<tr>
<td>2.5 F6: Measurement of Improvement Effects</td>
<td>23</td>
</tr>
</tbody>
</table>

### 3. Contribution of the Thesis

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Contribution Statement</td>
<td>28</td>
</tr>
</tbody>
</table>

### 4. Research Questions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
</table>

### 5. Research Approach and Methodology

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Research Paradigms</td>
<td>31</td>
</tr>
<tr>
<td>5.2 Empirical Research Design and Research Contribution</td>
<td>33</td>
</tr>
<tr>
<td>5.3 Research Setting</td>
<td>37</td>
</tr>
<tr>
<td>5.4 Categorization of Chapters</td>
<td>37</td>
</tr>
<tr>
<td>5.5 Research Approach</td>
<td>39</td>
</tr>
</tbody>
</table>

### 6. Conclusions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Validity Evaluation</td>
<td>42</td>
</tr>
<tr>
<td>6.2 Future Research</td>
<td>44</td>
</tr>
</tbody>
</table>

### Chapter 2 - Evaluating Evidence of Industry Application and Empirical Validation of Domain Analysis Solutions for Software Product Lines

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>60</td>
</tr>
<tr>
<td>2. Background and Related Work</td>
<td>61</td>
</tr>
</tbody>
</table>

### 3. Design

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Research Questions and Definitions</td>
<td>63</td>
</tr>
<tr>
<td>3.2 Search Strategy Development</td>
<td>64</td>
</tr>
<tr>
<td>3.3 Review Design</td>
<td>65</td>
</tr>
<tr>
<td>3.4 Validity Evaluation</td>
<td>70</td>
</tr>
</tbody>
</table>

### 4. Results and Analysis

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Included Studies Overview</td>
<td>71</td>
</tr>
<tr>
<td>4.2 Analysis</td>
<td>72</td>
</tr>
</tbody>
</table>

### 5. Conclusion

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
</table>

### Chapter 3 - Evaluating Alignment between Stakeholders

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>92</td>
</tr>
<tr>
<td>2. Background and Related Work</td>
<td>94</td>
</tr>
<tr>
<td>2.1 Product Strategy</td>
<td>95</td>
</tr>
<tr>
<td>2.2 MERTS</td>
<td>97</td>
</tr>
<tr>
<td>2.3 VMOST Analysis</td>
<td>100</td>
</tr>
<tr>
<td>2.4 Research Questions</td>
<td>100</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>3. MASS</td>
<td>101</td>
</tr>
<tr>
<td>3.1 Step 1 – Design Evaluation Framework</td>
<td>101</td>
</tr>
<tr>
<td>3.2 Step 2 – Identify SCS</td>
<td>105</td>
</tr>
<tr>
<td>3.3 Step 3 – Perform Evaluation</td>
<td>106</td>
</tr>
<tr>
<td>3.4 Step 4 – Perform Analysis</td>
<td>106</td>
</tr>
<tr>
<td>3.5 Step 5 – Conduct Follow-up Workshop</td>
<td>107</td>
</tr>
<tr>
<td>4. MASS at Ericsson – A Case Study</td>
<td>110</td>
</tr>
<tr>
<td>4.1 Execution</td>
<td>111</td>
</tr>
<tr>
<td>4.2 Validity Threats</td>
<td>118</td>
</tr>
<tr>
<td>4.3 Case Study Conclusions</td>
<td>119</td>
</tr>
<tr>
<td>5. Conclusions and Future Research</td>
<td>120</td>
</tr>
</tbody>
</table>

### Chapter 4 - A Homogenous and Consolidated View of Software Value

<table>
<thead>
<tr>
<th>1. Introduction</th>
<th>126</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Background and Related Work</td>
<td>127</td>
</tr>
<tr>
<td>2.1 The Absence of a Consolidated View on Value</td>
<td>127</td>
</tr>
<tr>
<td>2.2 Measurement and Valuation Methods</td>
<td>128</td>
</tr>
<tr>
<td>3. Research Methodology</td>
<td>131</td>
</tr>
<tr>
<td>3.1 Step 2: Literature Review Methodology</td>
<td>131</td>
</tr>
<tr>
<td>3.2 Threats to Validity</td>
<td>134</td>
</tr>
<tr>
<td>4. Literature Review Results</td>
<td>135</td>
</tr>
<tr>
<td>5. Step 3: Software Value Map</td>
<td>138</td>
</tr>
<tr>
<td>5.1 Taxonomy</td>
<td>138</td>
</tr>
<tr>
<td>5.2 Structure and Definitions</td>
<td>139</td>
</tr>
<tr>
<td>5.3 Interrelationships</td>
<td>142</td>
</tr>
<tr>
<td>6.1 The Concept of Impact Evaluation Patterns</td>
<td>144</td>
</tr>
<tr>
<td>6.2 Step 4.1: Pattern Creation Process</td>
<td>145</td>
</tr>
<tr>
<td>6.3 Step 4.2: Using the Patterns</td>
<td>150</td>
</tr>
<tr>
<td>6.4 Results and Lessons Learned</td>
<td>156</td>
</tr>
<tr>
<td>7. Conclusion</td>
<td>158</td>
</tr>
</tbody>
</table>

### Chapter 5 - Value Stream Mapping Framework for Software Engineering

<table>
<thead>
<tr>
<th>1. Introduction</th>
<th>166</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Background and Related Work</td>
<td>167</td>
</tr>
<tr>
<td>3. Software Value Map</td>
<td>168</td>
</tr>
<tr>
<td>4. Value Stream Mapping Framework</td>
<td>170</td>
</tr>
<tr>
<td>4.1 S1: Getting Started</td>
<td>172</td>
</tr>
<tr>
<td>4.2 S2: Mapping the Current Value Stream Map</td>
<td>177</td>
</tr>
<tr>
<td>4.3 S3: Identifying the Waste</td>
<td>183</td>
</tr>
<tr>
<td>4.4 S4: Improve the Process</td>
<td>187</td>
</tr>
<tr>
<td>4.5 S5: Perform Retrospective Analysis</td>
<td>189</td>
</tr>
<tr>
<td>5. Research Design</td>
<td>190</td>
</tr>
<tr>
<td>5.1 Case Description</td>
<td>190</td>
</tr>
<tr>
<td>5.2 Research question</td>
<td>192</td>
</tr>
<tr>
<td>5.3 Data Collection</td>
<td>192</td>
</tr>
<tr>
<td>5.4 Data Analysis</td>
<td>193</td>
</tr>
<tr>
<td>5.5 Validity Threats</td>
<td>193</td>
</tr>
<tr>
<td>6. Case Study Results</td>
<td>194</td>
</tr>
</tbody>
</table>
6.1 Instantiation of the framework 194
6.2 Outcome of framework application 201
7. Discussion and Lessons Learned 214
8. Conclusion 217

Chapter 6 - Innovative Features Selection using Real Options Theory 225
1. Introduction 226
2. Background 226
3. Real Options for Making Decisions about Innovative Features 228
4. Summary 230
Chapter 1 - Introduction

“There's an alternative. There's always a third way, and it's not a combination of the other two ways. It's a different way”
David Carradine

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. Traditionally, software engineering has been viewed and practiced in a value-neutral context [1-4]. However, current emerging trends require software engineering in general, and software product management in particular, to be viewed from a value-based perspective to ensure strategic, business-oriented and interdisciplinary perspectives, of which the technical perspective is but one central view [5-9].

Companies developing software intensive products risk rework, lost market opportunity and market failure due to shortcomings in integrating value-based decision-making into the management of product development [10]. A value-based decision is one that enables the planning and achievement of product cost, schedule, and quality while taking benefit, value and risk factors into consideration [11]. A value-based decision-making perspective is the fundamental foundation for any established business (software or otherwise), as it aids in binding a value-based context in which rational tactical decisions can be taken [12, 13]. Software engineering decision support plays a pivotal role in the value generation processes, as it assists professionals in taking the right strategic decisions and developing the right products [14]. Hence, decision support is a crucial component in achieving the goal of delivering value to the involved stakeholders [12]. The processes of determining goals, functions and constraints of a software system are termed as requirements engineering [15-17]. In a market-driven software intensive product development context, software product management is responsible for taking product management and requirements engineering decisions [18, 19].

One of the key elements in making the right decision is to decide and agree on a strategy and a set of value considerations for the development and evolution of a product [20-22]. However, practitioners, in product management and development, have different definitions and understanding of value aspects, and they do not consider all the value aspects while making decisions [23]. Decision-making, on the other hand, often involves deliberations utilizing different perspectives. The involvement of several distinct perspectives offers the opportunity for evaluating a decision using different priorities, however; this gives rise to the possibility for misinterpretations and misunderstandings in relation to the value aspects themselves, but also in relation to what value aspects to utilize in the decision process.

An aspect complicating things further is the difficulty in comparing different value aspects, as many are hard to measure, and if measures exist, they are not comparable. For example, how do you measure and compare architectural degradation to user experience?

The work presented in this thesis revolves around improving decision support, but also concretely defining and breaking down the concept of value as it pertains to the development of software intensive products. The definition of value was also extended and refined using empirical studies, and specific decision support tools1 were created to support practitioners in their daily work.

---

1 The word “tool” is used as a collective term denoting methods, models, frameworks and processes.
while enabling value to be used as a main criterion. Several tools were developed as part of this thesis. The results of the systematic review of domain analysis solutions can be used by practitioners to evaluate the usability and usefulness of solutions proposed for domain analysis for taking product line adoption/introduction decisions. The MASS method presented in this thesis can be used to evaluate strategic alignment and identify possible root causes for misalignment. To proactively strengthen strategic alignment, the Software Value Map and corresponding decision support material, can be used by product managers for making effective and efficient strategic decisions in relation to portfolios, products and process improvement, following a systematic and aligned process. Moreover, the use of real options theory can provide a richer option space for taking decisions about innovation candidates.

The individual contribution of each part of the thesis, and of each chapter, is presented in detail in the below (see Section 3).

**Thesis Contents and Disposition**

<table>
<thead>
<tr>
<th>Chapter 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Section 1: Introduction</strong></td>
</tr>
<tr>
<td>This section gives an introduction of the thesis.</td>
</tr>
<tr>
<td><strong>Section 2: Positioning the research</strong></td>
</tr>
<tr>
<td>This section positions the research by introducing software product management, decision-making concepts and theory of value-based software engineering. In this section, a brief introduction to software management and development activities in relation to the involvement of a software product manager is given. The challenges faced by a product manager while taking different decisions are also presented in this section.</td>
</tr>
<tr>
<td><strong>Section 3: Related Work</strong></td>
</tr>
<tr>
<td>This section briefly presents related work (detailed related work is presented in each respective chapter).</td>
</tr>
<tr>
<td><strong>Section 4: Contribution of the thesis</strong></td>
</tr>
<tr>
<td>This section highlights the contribution of the thesis chapter wise.</td>
</tr>
<tr>
<td><strong>Section 5: Research Questions</strong></td>
</tr>
<tr>
<td>Section 5 presents the research questions in order to address the challenges identified in Section 2.</td>
</tr>
<tr>
<td><strong>Section 6: Research Approach and Methodology</strong></td>
</tr>
<tr>
<td>This section outlines different research paradigms and the research approach and research methods utilized based on the chosen paradigm. This section also contains the research questions addressed, some basic theories behind the research methodology adopted, along with the research context in which the research was executed.</td>
</tr>
<tr>
<td><strong>Section 7: Conclusions</strong></td>
</tr>
<tr>
<td>Section 7 concludes this chapter along with the validity evaluation and future work.</td>
</tr>
</tbody>
</table>

**Chapter 2: Evaluating Evidence of Industry Application and Empirical Validation of Domain Analysis Solutions for Software Product Lines**

**Chapter 3: Evaluating Alignment between Stakeholders**

**Chapter 4: A Homogenous and Consolidated View of Software Value**

**Chapter 5: Value Stream Mapping Framework for Software Engineering**

**Chapter 6: Innovative Features Selection using Real Options Theory**
1. Positioning the Research
This section contains definitions and concepts that are needed to be defined and introduced for positioning the research presented in this thesis.

1.1 Definitions
A product can be defined as a good or service or a combination of the two that most closely meets the requirements of a particular market or segment and yield enough profit to justify its continued existence [24]. A software intensive product is one whose primary component is software [25]. Software intensive products vary in terms of degree of hardware (pure software vs. embedded systems), customization (generic vs. customized) and business model (packaged software vs. software as a service) [26]. A software product can be a part of a company's internal or external supply network.

A software product line (also referred to as a software product family) is a set of software-intensive systems/products that are sharing a common, managed set of features to satisfy the particular needs of a specific market segment and that are developed on a common platform (known as core assets) with defined (static or dynamic) [27, 28]. The purpose is to achieve production economies while increasing quality and decreasing cost [27, 28].

1.2 Software Product Management
Market-driven development is gaining increased interest in comparison to customer-specific system development [20, 29, 30], due to the emergence of markets for off-the-shelf or packaged product software [31, 32]. As a consequence, a shift in focus is required in the internal and external processes of a company. In such companies, over the years the role of software product manager has emerged to be of strategic value.

Traditional product management can be defined as a discipline, 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 (primarily in terms of revenue) to the business [18]. A product manager can be termed as a "mini CEO" representing an enterprise or business unit in strategy formulation and strategy realization through operational and tactical execution [18, 26]. The success of a product manager lies in balancing projects, people and politics. This role and the tasks associated with it have shown to be extremely complex [19]. The complexity arises from the need to exhibit abilities that range from technical savvy, to project manager and politician, all the time keeping an eye on the market and business aspects central for success.

Product management is a mature field and an established discipline; since Procter & Gamble introduced it in 1931 [33]. A number of books have been written on the subjects of product management and product marketing [34, 35]. In the nineties Cusomano [36], Kilipi [33] and Krishan [37] were the first to describe software product management as a separate business function. In the last decade, however, more and more professionals and researchers are paying more attention to this research domain, for example [38-43].

Although several of the existing product management practices [44, 45] can be applied to software product management, specific characteristics of market-driven software product management and development make product management of software intensive systems a challenging task. These characteristics are listed below:

1. Software products versus physical products: Software products are different from physical products as the manufacturing and distribution of extra copies of software do not require extra costs for the company [19, 46].
2. Ease of customization: Existing software 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 become highly complex [19, 20] (see Section 1.5.7 for corresponding challenges for software product management).

3. Number of stakeholders involved: The complexity of selection/rejection, organization and traceability of incoming requirements are further complicated by the fact that many internal and external stakeholders are involved (see Section 1.5.7 for corresponding challenge for software product management).

4. Release frequency: Due to the ease of making changes, and competitive markets, the software release frequency is relatively high in comparison to non-software products [19].

5. Overlaps in functionality: Due to overlaps in functionality of software products in a product line, the decisions related to choosing appropriate product portfolio management and domain scoping solutions are complex to make compared to non-software product (see Section 1.4.1 for corresponding challenge for software product management).

6. Iterative development by people: 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” run by people and continuously iterated. Whereas, 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 are handled separately [46]. The production and testing is mostly handled by the machines. Thus, in case of manufacturing, reduction in lead-times and costs are the most important aspects [47-49] whereas, in software development, in addition to these, other value considerations such as architectural value, human capital value and so on need to be evaluated.

These characteristics make managing and engineering a software intensive product a complex set of processes and activities in which a) constant decision-making is required using a criteria for evaluation of actions and outcomes, and b) almost always a group of stakeholders with different (and sometimes conflicting) perspectives is involved in the decision-making. Figure 1 shows different activities related to the management and development of software product (adapted from Xu and Brinkkemper [50], Jansen [51], van de Weerd et al. [19] and Kittlaus and Clough [26]). The activities have been categorized with respect to four distinct and fundamental perspectives namely: social, company, product and project [22].

Several of these activities are currently being researched. Examples include but are not limited to Terlouw et al. [52] on delivery, Dolstra [53] on deployment, Helms and Van Reijsen [54] on knowledge management, Lehtola and Kaappinen [55] and Berander [56] on requirements and architecture.

Since a product manager governs a product (or solution or service) from its inception and innovation to the end-of-life and decommissioning stage [18, 26], the responsibilities of a product manager are not constrained to one activity or to activities within one perspective. Rather he/she has a cross-organizational role, requiring a high degree of communication and coordination between all functional and organizational units [26]. A product manager collaborates with strategic management for market analysis, strategic product planning and roadmapping, product requirements engineering, portfolio management, innovation management [25], business case and costing, legal concerns and intellectual property rights. The product manager’s interface to development generally goes through project management, engineering management and project requirements engineering. He collaborates with marketing for product positioning, pricing, marketing mix optimization, marketing planning, product launch, opportunity management, and operational marketing. He interacts with sales and distribution for the product delivery model, customer relationship management, channel
preparation, sales management, and operational distribution. A product manager works together with evolution and services department to plan and coordinate technical, marketing, and sales support, services preparation, and services provisioning. Activities within process and quality such as knowledge management, software process improvement and quality management require input from the product manager. This clearly shows that a product manager constantly has to make decisions while interacting and coordinating with top management, marketing, sales, engineering, finance, quality and operations to make his products a business success [18, 22].

Figure 1: Process areas for Management and Development of Product Software (adapted from Xu and Brinkkemper [50], Jansen [51], van de Weerd et al. [19] and Kittlaus and Clough [26])

Figure 2 shows different decision-making scenarios within different process areas. The contribution of this thesis is related to decision-making in three important process areas in which software product managers actively take part, namely: portfolio management, product requirements engineering and processes and quality.

The research presented in this thesis is based on real needs in industry; therefore, inductive process assessment has been the starting point for every research contribution presented. This is deemed necessary for two reasons: 1) the research is based on the technology transfer model proposed by Gorschek et al. [57] which requires assessment, improvement and validation (this can be deduced from Section 5.5) and 2) a software product manager needs to have an assessment of the current process prior to the evaluation of the improvement potentials and commitment to their pilot test and application. Before going into detail of decision-making scenarios and challenges, Section 1.3 gives a brief introduction to decision-making theory, decision-making stages and decision-
making process. The decision-making scenarios and process areas that appear gray in Figure 2 are not addressed through the work presented in this thesis.

![Figure 2 – Decision-making scenarios within different process areas](image)

### 1.3 An Overview of Decision-making

#### 1.3.1 What is decision-making?

Decision-making can be regarded as the mental (cognitive) process resulting in the selection of a course of action among several alternative scenarios. Every decision-making process produces a final choice [58]. The output can be an action or an opinion of choice.

Human performance in decision-making has been the subject of active research from several perspectives. From a psychological perspective, it is essential to examine individual decisions in the context of a set of needs, preferences an individual has and values they search for. From a cognitive perspective, the decision-making process must be considered as a continuous process integrated in the interaction with the environment [59]. From a normative perspective, analysis of individual decisions is concerned with the logic of decision-making and rationality and the invariant choice it leads to [60].

Yet, from another perspective, it might be regarded as a problem solving activity which is concluded when an acceptable solution is reached. Therefore, decision-making is a reasoning or
emotional process which can be rational or irrational and can be based on explicit assumptions or tacit assumptions.

Logical decision-making is an essential part of all science-based professions, where specialists apply their knowledge in a given area to making informed decisions. For example, medical doctors often involve making diagnosis and selecting an appropriate treatment. However, it has been found through research, that in situations with higher time pressure, higher risks and uncertainties, experts use intuitive decision-making based on gut feeling rather than structured approaches. In such situations a recognition primed decision approach [61] is followed to fit a set of indicators into the expert’s experience and immediately arrive at a satisfactory course of action without weighing alternatives. Thus, based on the amount/certainty of the information available, the decisions can be categorized as (1) decisions under certainty and (2) decisions under uncertainty/risk.

1.3.2 Decision Processes
Theories of the stages of a decision process have been put forward by different researchers, for example Condorcet [62], Dewey [63], Simon [64] and Brim et al. [65]. However, the proposals by Dewey, Simon and Brim et al. are all sequential. Several authors, especially Witte [66], have criticized the idea that the decision process can, in general, be divided into consecutive stages. Rather he argues that the “stages” are performed in parallel [66]. Thus, a more realistic model should allow the various parts of the decision process to come in a different order in different decisions.

One of the most influential models that satisfy the above mentioned criterion was proposed by Mintzberg et al. [67]. According to the authors, the decision-making process consists of the following distinct phases, but these phases do not have a simple sequence.

Phase 1 – Identification: consists of two routines. The first routine is decision recognition routine in which problems and opportunities are identified in the streams of ambiguous, largely verbal data that decision makers receive. The second routine called diagnosis defined implies tapping of existing information channels and the opening of new ones to clarify and define the issues.

Phase 2 – Development: consists of two routines. The search routine that aims at finding the ready-made solutions and the design routine that aims at developing new solutions or modifying ready-made ones.

Phase 3 – Selection: consist of three routines. The first routine is screen routine, which is evoked only when a search is expected to generate more ready-made alternatives that can be intensively evaluated. The second routine is evaluation-choice routine in which the actual choice between the alternatives is made. It may include the use of one or more of three modes, namely: (intuitive) judgment, bargaining and analysis. In the third and last routine, authorization, approval for the solution selected is obtained higher up in the hierarchy.

The relation between these phases and routines is circular rather than linear. For example, a decision-maker may cycle between development and investigation to understand the problem to be solved.

1.3.3 Individual Decision-making
There are several steps involved in decision-making. These steps are shortly described in the subsections below.

1.3.3.1 Create decision-matrix
State decision problem as nXm matrix where n represents a finite set of possible alternatives that a decision maker can take and m represents a finite set of mutually exclusive states of nature that can result when actions are taken. Alternatives are typically courses of action that are open to a decision
maker at the time of decision. The set of alternatives can be open (meaning new alternatives can be invented or discovered by the decision maker) or closed (meaning no new alternatives can be added). In decision theory, the various unknown extraneous factors are summarized as cases and are called states of nature. The possible outcomes of a decision are defined as the combined effect of a chosen alternative and the state of nature that obtains.

1.3.3.2 Calculate Expected Utility
Expected utility can be defined as expected degree of satisfaction. Expected utility $E(a_i)$ can be calculated with the following formula:

$$E(a_i) = \sum (p(a_i)^k \times u_{ij}) \quad \text{eq 1}$$

where $p(a_i)$ is the decision maker’s personal probability and $u_{ij}$ is the utility acquired.

Personal probability is defined as degree of confidence that may be different with different persons. Utility can be described as the degree of satisfaction. A utility function describes a decision maker’s preferences between the different world states and it is used to assign a single number to express the desirability of a state [68].

Utility can be measured by:
1. Direct questioning
2. Indirect methods by using the device of binary lotteries [68]

1.3.3.3 Choose an Optimal Action
An optimal action can be defined as the one which maximizes expected utility. However, to compute the expected utility of all actions it is required to know the precise number of the probability $p_j$ for all states. This might be difficult in a real world situation. What, on the other hand, can be done is to try and find actions with lower expected utilities and exclude those from the list of actions through dominance. For example, if $E(a_k) > E(a_e)$ where $a_e$ denotes all actions in the set of actions except $a_k$ then

$$\text{Max } E(a_i) = E(a_k)$$

1.3.4 Group decision-making
Group decision-making is a situation faced when people think or are brought together to solve problems in the anticipation that they are more effective than individuals under the idea of synergy. In group decision-making, expected utilities are calculated for all the stakeholders involved in decision-making and an optimal action is chosen by the group rather than an individual.

There are several methods for group decision-making, for example:
1. Consensus decision-making aims to avoid "winners" and "losers". Consensus requires that a majority approves a given course of action, and the minority agrees to go along with the course of action [69]. In other words, if the minority opposes the course of action, consensus requires that the course of action is modified to remove objectionable features.
2. Voting-based methods, for example range voting lets each member score one or more of the available options. The option with the highest average is chosen. Majority requires support from more than 50% of the members of the group. Thus, the bar for action is lower than with unanimity and a group of “losers” is implicit to this rule. Plurality, where the largest block in a group decides, even if it falls short of a majority.
3. Delphi method is a structured communication technique for groups, originally developed for collaborative forecasting but has also been used for policy making [70].
4. Democracy is a facilitation method that relies on the use of special forms called Democracy Sheets to allow large groups to brainstorm collectively and recognize agreement on an unlimited number of ideas they have authored.

Until quite recently, decision-making in software engineering has been considered an individual activity by practicing “separation of concerns” [12]. For example, a software engineer was kept out of the requirements selection and prioritization decision-making scenarios, and his responsibility was just to convert the specified requirements into verified code. This led to software projects failures because different stakeholders’ utility functions or value propositions were not understood and evaluated [12]. In order to fill this void, Boehm and Jain [71] proposed a “4+1” theory of value-based software engineering (VBSE) which states that an enterprise will succeed if and only if it makes winners of its success critical stakeholders”.

1.3.5 A “4+1” Theory of Value-Based Software Engineering

In the center of the theory of value-based software engineering (VBSE) is the success-critical stakeholders (SCS) win-win Theory W [71], which addresses the questions of “what values are important?” and “how is success assured?” for a given software engineering enterprise. The four additional theories that it draws upon are utility theory (how important are the values), decision theory (how do stakeholders’ values determine decisions?), dependency theory (how do dependencies affect value realization), and control theory (how to adapt to change and control value realization).

Figure 3 – The “4+1” Theory of VBSE overall structure (from [71])

However, the above theorem does not describe how to achieve and maintain a win-win state. This requires a Win-Win Achievement Theorem: “making winners of your success-critical stakeholders requires (1) Identifying all of the success-critical stakeholders (SCSs); (2) Understanding how the SCSs want to win; (3) Having the SCSs negotiate a win-win set of product and process plans; (4) Controlling progress toward SCS win-win realization, including adaption to change”.

According to Boehm and Jain, identifying all of the SCSs is in the province of dependency theory. Dependency theory covers the full range of theories that help to reason and describe how dependencies affect value realization. These include theories about product dependencies such as
physics, computer science, and architectural theories [72, 73], theories about process dependencies, such as scheduling and concurrency theories, theories about stakeholders interdependencies, such as sociology and organization theories [74, 75] and theories about product, process and stakeholders interdependencies, such as economic, management, and system engineering theories [76-80]. A key technique that can be applied to identify all SCSs is the Result Chain [81].

Understanding how the SCSs want to win is in the region of utility theory [82, 83]. The utility theory deals with the analysis of choices among risky projects with (possibly multidimensional) outcomes. Utility theory is based on the utility functions of the SCSs. Misunderstanding SCS utility functions may not lead to failure of an organization if an organization happens to get lucky. However, understanding how the SCSs want to win is a necessary condition for achieving win-win [71]. Utility theory also has several branches. For example, the satisficing theory of bounded rationality [84], multi-attribute utility theory [85] and so on.

Having the SCSs negotiate win-win plans lies within decision theory. Decision theory also has many aspects, for example negotiation theory, game theory, multi-attribute decision theory, statistical decision theory and the buying of information to reduce risk, real options theory and theory of justice.

Controlling progress towards SCS win-win realization lies within the realm of control theory [71]. The necessary conditions for successful enterprise control are observability, predictability, controllability and stability [86]. From value-based decision-making perspective, it is fundamental to apply control theory principles to the expected value being realized by company, product and project rather than just to track and project progress with respect to plans [71]. Business value and mission value achievement monitoring and controlling systems such as adaptive control mechanisms (market watch and plan renegotiation), multi-criteria control mechanisms (for example, BTOPP [81] and balanced scorecards [87] [88]) should be used.

1.3.6 Criteria for decision-making
It is simpler and quicker to make decisions in a value-neutral setting, especially when short-term revenues and customer satisfaction (in terms of costs involved and functional and non-functional requirements implemented) are the major focus of management [89]. However, this simplicity is gained at the customer's expense that deliver products cannot provide the best possible outcome to the stakeholders [11].

VBSE is a framework for improving products delivered to customers by incorporating value considerations into technical and managerial decisions that are made during product management and development [11]. VBSE community argues that decisions in and about software management and engineering should focus on value created by the software product [90, 91]. However, incorporating different value considerations into decision-making will present challenges for both software engineers and software product managers [11]. The reason being, at a quick glance, value might appear as a single, unique criterion to be considered while making decisions. However, a deeper analysis of the value generated by product software reveals that it is a multi-dimensional concept in itself which involves different attributes (for example customer perceived value, project value, competitive advantage value, innovation value and so on [9, 92-95]), time periods (depending on life cycle of product and planned releases) and stakeholders (customer, product managers, sales and marketing, developer, testers, support group, suppliers and partners). A number of multi-criteria decision analysis methods are available such as aggregated indices randomization method (AIRM), analytic hierarchy process (AHP), analytic network process (ANP), data envelopment analysis, dominance-based rough set approach (DRSA), ELECTRE (Outranking), the evidential reasoning approach (ER), goal programming, Grey relational analysis (GRA), value engineering, value analysis, multi-attribute utility theory [96]. However, before stepping into such an analysis, a prerequisite is to consolidate all the criteria relevant from different perspectives for evaluation [97]. This is also central for understanding, i.e. establishing shared meaning for concepts (value aspects)
under consideration, as different people frequently understand the same term to represent different concepts [97, 98].

To summarize, the following aspects need to be considered for value-based decision-making:

- A1: Careful identification of all SCSs
- A2: Elicitation and reconciliation of stakeholders’ value propositions
- A3: Better understanding of links between technical design mechanisms, context and value creation to enable both better education and decision-making in any given situation
- A4: Consolidated view of value aspects relevant from different perspectives for decision-making
- A5: Richer option space
- A6: Dynamic monitoring and control mechanisms to better guide decision-makers through the options pace in search of value added over time

After a brief introduction to the process and theories of decision-making and value-based decision-making, Sections 1.4 - 1.6 detail process areas in which the SPM encounters different decision-making scenarios.

### 1.4 Portfolio Management

Professional product management is essentially a matter of well-organized processing of issues related to requirements, products and releases [19]. The scope of work of product management concerns the complete set of products for a company, the so-called product portfolio. Small or young companies may have a portfolio of just one product, whereas larger companies can have several products in a portfolio. Portfolio management is formally defined as the term used for “managing investment decisions over time following profit and risk criteria” [26]. Thus, portfolio management encompasses decision-making about the set of existing products, introducing new products by examining market trends and the product development strategy, making decisions about the product life cycle, and establishing partnerships and contracts [19, 26]. According to van de Weerd et al., software product line management also lies in this area [19].

Successful portfolio management is about achieving a balance between the four potentially conflicting goals of 1) maximizing the financial value of the portfolio, 2) linking the portfolio to strategy, 3) balancing it on relevant dimensions, and 4) ensuring that the total number of ongoing activities is possible [99]. Different portfolio management techniques, for example financial and economic models, scoring models and mapping approaches prioritize these goals differently [100, 101].

According to state-of-the-art, there are two basic alternatives for implementing the portfolio management process in practice. The first one proposes making decisions through in-depth reviews of each ongoing product/project and managing the portfolio in a bottom-up manner (e.g. gates dominate [101] and model I funnel [34]). The second is top-down, where decisions are based on the portfolio as a whole view (e.g. portfolio reviews dominate [101] and model II funnel [34]).

The first alternative is appropriate for larger organizations in mature businesses with dedicated resources and fairly static and stable portfolios, where the emphasis is more on making sound go/kill decisions for individual products/projects than re-prioritizing the entire portfolio after a period of time. On the other hand, the second, is better suited for smaller organizations in fast-paced and fluid markets as it allows for more dynamic resource allocation by periodic reviewing of the entire portfolio [34, 101].
1.4.1 Software Product Lines and Domain Analysis
A number of case studies have shown the introducing/adopting product line approach improves organizational performance. A first step towards adopting/introducing a product line approach is to choose an appropriate domain analysis solution to capture the features. 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 [102]. Domains are areas of expertise that can be used for creation of a system or a set of systems [27]. 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 [27]. It helps in the identification of the specification of the systems in the product line. It involves various activities that can be categorized as modeling and scoping [102]. Modeling is defined as capturing information and organizing it into a model whereas, scoping is defined as a decision-making activity.

1.4.1.1 Domain Modeling
In the modeling category, the activities identified are [102]:

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 contain 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.

1.4.1.2 Domain Scoping
In the scoping category, the activities are:

1. Domain scoping is the process of identifying boundaries for a domain which are appropriate for implementing systems in the product line [28].
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 [27]. The result of a scoping activity is a scope definition document which becomes a product line core asset [27]. 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 [27].

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 [102]. 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.

1.4.2 C1: Identification of appropriate domain analysis solutions
A product manager carefully needs to analyze and evaluate available domain analysis solutions and select the one that appears to be a better alternative. Many domain analysis solutions have been presented over the years [103-106]. However, presently SPMs in the industry are faced with the challenge of identifying an appropriate one, and this is complicated by the fact that evidence of usability and usefulness of the proposed solutions is rare [107-109]. Thus, one of the decision-making challenges faced by software product managers is the identification of appropriate domain analysis solutions.

1.5 An Overview of Product/Market-driven Requirements Engineering
In contrast to traditional requirements engineering, requirements in market-driven requirements engineering (MDRE) for software products 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]. This influences all different phases of MDRE including elicitation practices, analysis and management. Figure 4 gives an overview of a “generic” MDRE process based on several sources in literature [29, 110-114]. Each part of the process is described in sections below.
1.5.1 Requirements Elicitation

The initiation of an MDRE effort is not directed by an external request or a customer order, rather it is continuous in nature. The development efforts in this context are initiated as needed per release plan (following a product or product-technology roadmap). To simplify, it can be stated that market(s) is the customer. Certain entities in the market can be identified. For example, key-customers are a part of the market; however, due to their importance they usually have the possibility to put forward feature requests directly to the developing organization in addition to going through defined interfaces like marketing, sales or support. Figure 4 shows additional potential external sources of features; distributors, suppliers, partners that are part of the complete value chain. This is getting more and more common as software is becoming part of products like cars, robots and services like IPTV solutions. An example could be development organization providing pre-paid/post-paid solutions to mobile phone operators. In this case, the market consists of all mobile operators, the end-users using the services, content providers and third parties providing products or services to be used with the products/services offered by the development company.

The different elicitation techniques which can be used in market-driven context have been proposed, e.g. focus groups, workshops with key customers, customer working groups [116], surveys. Although all the techniques can provide better insight into customers’ values and priorities and can help to tap into new possibilities through a new set of features, an obvious downside is that this can easily result into multiple and conflicting preferences between different customers and between customers and development organization [116].

On the other hand, many requirements come from internal sources, for example architects, testers, sales and support team. These can be termed as innovations as they are often an attempt to add new and unique features and/or characteristics to a product or are proposals of unique ways to bundle products and/or services as solutions. Extensive domain knowledge resides with the
development organizations which if used correctly can result in profitable innovative ideas providing
the competitive edge [45].

1.5.2 Requirements Analysis
All requirements are stored in a central repository independent of the source of origin and
subsequently analyzed. This first involves performing early acceptance/rejection of the incoming
requirements (often called triage) [117]. For a product manager; it is a challenge to handle large
amounts of requirements continuously, performing early triage, selecting the ones aligned with a
specific product’s goals and strategy [20, 118].

Following early triage analysis is performed where implementation costs and resources are
estimated for requirements prioritization. Having as accurate estimates as possible directly affects
the ability and quality of requirements prioritization and release planning [119, 120]. Function points
or feature points can be used to estimate costs and resources based on requirements. However, the
use of non-structured informal techniques for requirements estimation seems to be pre-dominant,
where practitioners rely on expert judgment or gut feeling [121, 122].

An important aspect to be investigated during analysis is requirements interdependencies, which
can influence requirements selection [123]. In the market-driven context, value-based dependencies
that are directly related to customer value and development cost seem to be pre-dominant [124].
According to Carlshamre et al. [124], only about 20% of all requirements are relatively singular,
making dependencies a highly relevant aspect in requirements analysis.

1.5.3 Requirements Prioritization
The quality of a software product is often determined by the ability to satisfy the needs of customers
and users [125]. Hence, eliciting and specifying the correct requirements and planning suitable
releases with the right feature set is a major step towards the success of a project or product. If
wrong requirements are implemented and users resist using the product, it does not matter how solid
the product is or how thoroughly it has been tested.

In market-driven context, there is a continuous stream of incoming candidate requirements than
can be realized within the time and cost constraints. Thus, the challenge is to select the ‘right’
requirements out of a given superset of candidate requirements so that all the different key interests,
technical constraints (e.g. architecture sustainability), business strategic aspects (e.g. focusing on
new-market segments) and preferences of the success-critical stakeholders are fulfilled, and the
overall business value of the product is maximized” [126]. Therefore, requirements prioritization is
an important component in the value generation process when developing software products.

Several methods for achieving requirements priorities exist; AHP [127], the 100-point method
[128], attainment [129], top ten requirements [130] and the planning game [131] to name a few.
However, during prioritization in value-neutral settings, mostly estimated costs and project timelines
are used with some vague notion of importance of requirements. While models for cost estimation
are well established, definitions of value considerations (supporting decision-making and
prioritization) and frameworks are missing [132]. When the aim is to decrease time-to-market with
maximum stakeholder satisfaction, a value-based approach is necessary, and decision support for
release planning (and hence prioritization) can play a key role in identifying value propositions [92].
This is necessary to identify the 20% of the features that generate 80% of the software value [91].

1.5.4 Requirements Selection and Release Planning
After initial triage, requirements are analyzed (estimated, prioritized, and crucial dependencies
mapped) and the actual selection and release allocation take place. Product roadmaps (see Figure 4)
can be used as an approach to document and communicate plans for future releases [133-135]. There
are many types of roadmaps described in literature, in fact, any forward looking document is called
roadmap [136] which has the purpose of “mapping” and aligning efforts and resources towards a common goal. A road map needs to reveal several aspects, for example, themes of a certain product release (a theme could be offering certain functionality, concentrating on improving quality and so on), restrictions (e.g. what are the restrictions in terms of risk, time, internal technical issues and resources available) and milestones (planned releases with respect to the strategic goals).

A road map can be seen as an explicit concretization of product strategies depicting the long-term plans of a product. It is a relatively common way of representing long-term targets based on development in the context of time and releases [136]. Product strategies are in turn a concretization of company’s and business strategies pertaining to an individual product [137].

Ignoring product strategies (by only looking at short-term priorities) may mean that a product is successful in short-term, but at the expenses of the long-term goals [138, 139]. For example, if availability requirements are traded-off with performance requirements, when the long-term plan for the product is eventually to break into new market segments where availability is the most crucial, can eventually result is revenue decrease. 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 [20-22, 138, 140]. Thus, a fundamental aspect to be considered while formulating product strategies is the balance between business perspective and technical perspective. There is always a risk that one perspective dominates the other, predominantly the business one [141].

It is extremely important to note that requirements selection within the boundaries of a product strategy following a roadmap, however, does not guarantee success, as the strategies can be flawed. However, having up-to-date product strategies, which reflect all the fundamental knowledge – from both technical and marketing perspectives and lessons learned from decisions made early when the consequences and results are available [142] – will most certainly increase the chances of developing a successful product [138, 139].

In order to facilitate release planning decision-making, there are several release planning solutions e.g. REPEAT process [143] and EVOLVE [92], cost-value approach [144], incremental funding method [145] and F-EVOLVE* [146]. However, all these solutions require hard to calculate parameters such as relative accurate estimates, priorities, guiding strategies and clear release dates. Moreover, in value-based release planning methods [144] [145] [146], value is mostly estimated as revenues generated and market value for customers without considering other tangible and intangible value aspects (such intellectual capital value, innovation value and so on).

1.5.5 Requirements Validation
Traditional requirements validation is performed in close cooperation with the customer (in case of bespoke RE); however, in MDRE this is complicated for obvious reasons. There is a possibility to cooperate with key customers but their input may be hard to obtain, and in addition, they may not be representative of all requirements implemented in a release [119, 147]. This is further complicated by the fact that some requirements are invented to suit an imaginary customer [111]. In market-driven context, validation can be performed internally using the knowledge of the development organization. Another method for requirements validation can be the use of beta releases of upcoming releases, and having a subset of real customers who test the product.

1.5.6 Requirements Management
The requirements repository (for example an RE tool or database with a front-end) can act as a central storage for all requirements. It is important to consider scalability, traceability and distributed work related aspects while selecting an appropriate specification tool.
The use of a tool also enables the use of attributes. Attributes are a good way of structuring requirements in a repository. Examples of attributes can be ID (unique identifier), Title, Description, Rationale, State (what is the state of requirement: new, dismissed, specified, planned for release, released and so on [112, 143], source, dependency etc. Different development organizations may need different attributes [112].

Requirements change in MDRE and even more so compared to bespoke RE because of new high priority requirements, market change. In MDRE changes to requirements prior to release allocation can occur due to the emergence of new high-priority requirements, market changes can lower the priority of an already prioritized requirement and so on. Changes can be handled through general change management processes. However, change management is complicated if changes occur after a requirement has been allocated to a release. Changes may require re-prioritization, re-selection and re-allocation of all requirements to a certain release in the worst case, due to e.g. dependencies and urgencies for release.

Change management can be performed by e.g. a steering committee [119] or through the use of a traditional Change Control Board (CCB) [148]. An important constraint to remember is that the time-to-market (release dates) is generally fixed; thus the selection of requirements for a release has to adapt to this fact [119, 148]. In MDRE, time aspects are even crucial enough to be prioritized over e.g. quality aspects [149].

Since, in MDRE context, requirements are not put forward by a single customer rather the selection decisions are largely made by the stakeholders internal to the company keeping in view customer (segments) preferences, market trends, competitors and so on, there is a need to analyze the quality of selection decisions post-fact when the results of the decisions are available. For this, it is required to store the evaluation of selected requirements and decision rationale for both selected and rejected requirement. However, this is often missing in practice [150, 151].

### 1.5.7 MDRE – Challenges

With respect to software engineering, insufficient requirements engineering is typically the first sign of product management failure [22]. In the context of market-driven requirements engineering for software intensive product management, a product manager is faced with several decision-making challenges in different processes. Several MDRE related challenges based on industry experiences are reported in literature [19, 29, 30, 110-112, 119, 124, 149, 152-155]. Some of the challenges (relevant to the research presented in this thesis) based on literature and the author’s personal industrial experience [23, 118] are summarized below.

#### 1.5.7.1 C2: 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 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 [19]. 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 result 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 [20-22, 118]. Thus, from a value creation perspective it is vital 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 [21, 22, 118, 141].

1.5.7.2 **C3: Alignment with respect to Product Strategy**
Requirements can be viewed as the least common denominator on which decisions regarding what to include in a product’s release/offering are made. As discussed in the previous challenge (C2) 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 values, 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 and values important and relevant for software product management and development decisions.

1.5.7.3 **C4: Selection of Innovative Features**
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. However, the challenge is how to select most promising innovative requirements given the inherent risks and uncertainties involved.

1.6 Processes and Quality
As global competitiveness and shorter time-to-market come to the software development industry, the search is on for a better way to create first-class software rapidly, repeatedly, and reliably. Thus, the significance of software process quality has markedly increased, and development organizations have made it one of their main business objectives [156].

As stated by Rainer and Hall in [157], many software projects fail due to poorly designed and managed software processes. The results of the CHAOS report (of 2009) showed a remarkably low number of successful projects (32% of all projects) which were delivered on time, on budget, with required features and functions [158]. According to the chairman of the Standish group, 44% were challenged because they were late, over budget, and/or with less than the required features and functions and 24% failed which were cancelled prior to completion or delivered and never used. This situation demands increased focus on the software development and management processes.

1.6.1 **Software Process Improvement Initiatives**
Software process improvement (SPI) is an important research field within Software Engineering. It helps practitioners achieve better results in terms of quality in their software projects [159]. There exist several well-known and established SPI framework used for process assessment and improvement e.g. Shewart-Deming’s PDCA (Plan-Do-Check-Act) [160], Basili’s QIP [161], CMM (capability maturity model) [162, 163]. Most of them are based on a general principle of four straight forwards steps: “evaluation of the current situation”, “plan for improvement”, “implement the improvement” and “evaluate the effect of improvements”.

Page 18 of 231
An elementary division of SPI frameworks can be done based on whether or not they are bottom-up (inductive) [155, 164] or top-down (prescriptive) model in nature. Below inductive and prescriptive SPI frameworks are characterized.

### 1.6.2 Prescriptive (model) based frameworks

A prescriptive approach is based on a collection of best practices describing how e.g. software should be developed. These models are termed as prescriptive because they prescribe one set of practices to be adhered to by all organizations. No special consideration is given to an organization’s unique situation or needs other than how the development process (at the organization subject to SPI) is in comparison to the one offered by the framework [165, 166]. The tools of assessment for such models are designed towards benchmarking against the set of practices prescribed by the model.

#### 1.6.2.1 Capability Maturity Model

The Software Engineering Institute’s CMM (capability maturity model) [162, 163] is one of the most well-known model-base SPI standards. It is based on standardizing the contents of processes according to a pre-defined number of practices. It starts with assessment of the organizational maturity (benchmarking against CMM practices). The process assessments generically associated with CMM are called CMM-based appraisals (CBAs). Subsequent to the assessment is planning and preparation for introducing relevant practices to conform to CMM. This includes both what practices are used, and in what order they are implemented. After the execution of the improvement, a new assessment is made to evaluate the success of the improvement as well as prepare for the next interaction [167].

The practices of CMM are organized into Key Process Areas (KPA), e.g. Requirements Management and Software Quality Assurance. Each KPA is placed on a certain level of maturity ranging from 1 to 5. The maturity levels are shown in Figure 5 where the bottom most level (1: Initial) is the lowest level of maturity where practices are undefined, the process is ad-hoc and non-repeatable (CMM does not have any KPAs on this level). As the maturity of an organization increases (Level 2 and up), KPAs (with relevant practices) are introduced for every step in a re-defined manner.

Figure 5 - Maturity levels in CMM
The original CMM is rather general in nature and was not intended for any specific discipline (e.g., software development), a number of specific CMM versions emerged, e.g., SW-CMM (CMM for software) and IPD-CMM (Integrated Product Development CMM), CMMI [168], ISO/IEC 15505 (a.k.a SPICE – Software Process Improvement and Capability dEtermination).

1.6.3 Inductive Frameworks
Basili’s QIP is based on bottom-up approach that is inductive in nature, i.e. what is to be performed in terms of improvements is based on a thorough understanding of the current situation (processes) [161, 169]. QIP proposes a tailoring of solutions based on the critical issues identified in the project organization, and before actual implementation, the solutions are evaluated in pilot projects before an official change is made to the process [170].

Lean engineering process improvements such as Value Stream Mapping can also be classified as inductive frameworks. Value stream mapping is used to apply lean thinking for eliminating wastes and improving cycle time and quality in engineering. A value stream contains all the tasks and actions (both value added and non-value added) currently required to bring a product through the main process steps to the customer a.k.a end-to-end flow of process. The lean philosophy emphasizes to optimize the whole value stream, from the time it receives an order to address a customer until the software is deployed and the need is addressed because if a company focuses on optimizing something less than the entire value stream, the overall value stream is guaranteed to suffer [47].

For inductive framework, there is no general initial assessment like performing baselining against a pre-defined set of improvements rather the experiences from existing processes in projects are used to base improvements on [171]. The improvements can be in the form of new process, methods, techniques and/or tools.

1.6.4 General SPI – Success Factors
There is no shortage of SPI experiences reported from industry assessment and improvement efforts, based on both inductive and prescriptive frameworks, like the ones described in Sections 1.6.2 and 1.6.3. Several success factors have been identified during the practical application of these frameworks as determinants of success [172-174]. Some of these factors (relevant to the research presented in this thesis) are summarized below, and to some extent exemplified with the use of frameworks described in Sections 1.6.2 and 1.6.3.

1.6.4.1 SPI Initiation Threshold
The initial critical success factor is, of course, commitment to initiate SPI. The threshold for initiating and committing to an SPI effort is often high due to the associated resources that have to be committed. An assessment-improvement cycle is often very expensive and time-consuming [175]. For example, a typical SPI cycle using CMM can take anything from 18 to 24 months to complete and demands a lot of resources in order to be successful [165].

Additionally, the threshold is not lowered by the fact that many perceive extensive SPI frameworks, e.g. CMM and SPICE as too large and bulky to get an overview of and to implement [176, 177]. This is particularly a serious concern for small and medium-sized enterprises (SMEs) where dedicating time and resources for assessment and improvement is always an issue [176, 177]. In order to reduce the size and complexity of software process improvement initiatives, light SPI frameworks have been proposed e.g. IMPACT project [178], IDEAL [179] and Dynamic CMM [180].

Lean initiatives in manufacturing, logistics, services, and product development have led to dramatic improvements in cost, quality and delivery time [181]. However, contrary to the traditional manufacturing, software development is a complex activity involving direct and indirect value
considerations relevant from different perspectives. Therefore, it is fundamental to adapt existing lean approaches (for example value stream mapping) to software engineering context.

1.6.4.2 Commitment and involvement

Assuming there is a genuine desire and need for SPI in an organization there has to be a strong commitment from management, which is considered as one of the most central factors for SPI to be successful. SPI efforts need to be actively supported, and management needs to dedicate resources for SPI effort. An example of re-occurring problem is an assumption that SPI work can be accomplished along with the organization’s current workload [159, 182-184]. Management commitment is, of course to a greater extent, associated with cost and resource issues presented above, as management is less likely to commit to an SPI effort if it is very costly and time consuming.

However, commitment from management is not enough to ensure success. There has to be commitment and involvement by management, middle management and the development staff (i.e. stakeholders that are involved in SPI work and/or that are affected by the outcome). It is a genuinely good idea to let the people (doers) working with the processes every day actively participate in the improvement work [159, 182-187] as they generally have a good understanding and insights about which areas need improvement, and this knowledge often becomes explicit during an assessment activity [188]. Further, representation of doers involved in all the activities for delivering a product ensures improvement with an end-to-end perspective.

1.6.4.3 Goals and Measurement

Prior to any SPI initiative there needs to be a clear and well defined goals pertaining to the SPI activity. This is achieved through proper preparation and planning, but also having a clear focus on what needs to be done (what is important to improve and why) [159, 182-184].

A crucial part of any SPI activity is the ability to measure improvement effect in order to identify whether or not an improvement is successful [159, 189, 190]. Prescriptive model-based SPI frameworks measure e.g. adherence to a set of pre-defined practices primarily, although measurement methods can also be used as a part of this. Whereas, inductive framework e.g. QIP propagates strict adherence to the collection and analysis of metrics through e.g. GQM in order to ascertain if an improvement has been successful (in comparison to the goals set up). In value-based context, impact of improvements on value aspects derived from goals needs to be measured and analyzed because a positive effect of an improvement on one value aspect (say lead-time) might actually have a detrimental effect on other value aspects (e.g. product architecture sustainability).

1.6.4.4 Summary of General SPI – success factors

Factors discussed in Sections 1.6.4.1-1.6.4.3 are summarized below:

- SPI Initiation Threshold can be decomposed into two main factors:
  - F1: Time to return-on-investment
  - F2: Cost/Resources/Size
- Commitment and Involvement can be broken down into
  - F3: Commitment and involvement to SPI by management
  - F4: Commitment to SPI and Involvement in the SPI initiative by process doers (e.g. technical managers, developers, testers)
- Goals and Measurement
  - F5: Vision (of the SPI effort with clear and well-defined goals)
F6: Measurement of improvement effects (considering all value aspects derived from the defined goals).

Although this is not a direct challenge of a product manager, it is important for a product manager, as the owner and responsible for the product, to carefully analyze the suggested improvements to a product development process based on the above mentioned factors. The formulation of F1 through F6 is based on experiences from general SPI efforts undertaken in industry. The factors are used to discuss thesis contribution in Section 3.

2. Related Work

From the perspective of the challenges presented in Sections 1.4 - 1.6 and SPI success factors given in 1.6.4.4, it is central to investigate related work that has been done in order to elaborate on, or 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.

2.1 C1: Identification of Appropriate Domain Analysis Solutions

A number of studies [191-197] 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 the challenge C1 with the focus on usability and usefulness aspects.

2.2 C2: Creation of Product Value

Several researchers have presented value components and corresponding valuation/measurement solutions needed for making decisions about software product development [9, 95, 198-201]. However, the contributions are often isolated and have a limited perspective, focusing on e.g. only cost, or only product characteristics like usability. Consequently, a complete picture of value components, relevant from different perspectives required for taking software product management and development decisions, is missing. For example, from the product perspective, if software value is measured by measuring external and internal quality attributes [202], say usability or reliability, it is only one isolated measure. The usability value is often shown as a number (for example, understandability = A / B, where A = Number of functions (or types of functions) understood, and B = Total number of functions (or types of functions)). The question remains, however, how does this number relate to the software product value relevant from different perspectives, like the overall customer perceived value, or the impact on internal business processes?

Similarly, cost-benefit analysis methods like CBAM [203] and LiVASAE [1] based on subjective evaluation of architectural strategies with respect to quality attributes of the architecture, fall short on connecting cost-benefit discussions to a broader context (e.g. customer perceived value, innovation value and or the impact on internal business processes).

Business research investigates value components from a number of other perspectives, i.e., production value, differentiation value, intellectual capital value and shareholders’ value [198, 204]. However, in software engineering literature, these have little or no explicit connection to the software product planning (requirements triage, requirements selection and product management), and/or development (design, coding and testing) activities [91, 205]. It is central to realize that software related decisions cannot be extricated from business value concerns in a market-driven software product development context. Thus, for creation of product value (Challenge C2), a consolidated view of all value aspects relevant from different perspectives is missing in the literature.
2.3 C3: Alignment with respect to Product Strategy
A number of studies have suggested methods/tools/techniques for achieving, assessing and maintaining alignment [206-213], however, none of the studies have assessed/evaluated alignment among involved stakeholders to aid in decisions related to value creation in the context of market-driven software intensive product development (Challenge C3). 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 the product and company perspectives [22]. This ensures that there is a common understanding of the company’s goals and objectives for a particular product.

2.4 C4: Selection of Innovative Features
Sharkawy and Schmid [214] have proposed a heuristic approach for supporting product innovation in requirements engineering. The approach uses semantic-based technologies to derive new idea triggers. On the other hand, Regnell et al. [215] have proposed a measurement framework focusing on four areas: innovation elicitation, selection, impact and ways-of-working. While Racheva et al. [216] have discussed the application of real options for requirements prioritization in the projects’ context, there has been no explicit research on the use of real options for selection of innovative features. Thus, there is a need to investigate possibilities that can incorporate inherent risks and uncertainties in the selection of innovative features.

2.5 F6: Measurement of Improvement Effects
With respect to software process improvement, Toyota revolutionized the automobile industry with their approach of “Lean Manufacturing” in the 1980s. Womack and Jones [217], through publications enhanced the original concept of lean manufacturing into the framework of “Lean Thinking” that gained popularity during the 1990s. Womack and Jones distilled the essence of lean approach into five key principles demonstrating how the lean approach and relevant practices can be extended beyond the automotive industry to any company and organization in any sector. In the early 2000s, Poppendieck and Poppendieck [218] further transferred these principles and practices from the manufacturing domain to specifically suit the software development context and called it “Lean Software Development”.

Rother and Shook [219] provide a detailed account on the practical implementation of Value Stream Mapping, which is also used in other sectors and services related industries. The “Lean Advancements Initiative” research consortium at MIT, has for many years applied lean concepts for improving aerospace product development processes. Their main work “Product Development Value Stream Mapping Manual” [220] offers extensive academic research for learning. Pavnaskar and Gershenson [221] have discussed some differences between manufacturing and engineering and proposed a set of icons, definition of process data and added representation of parallel and sequencing processing. However, explicit strategies as to perform each step of Value Stream Mapping are missing both in MIT manual and Pavnaskar and Gershenson work.

Mujtaba et al. [222] applied Value Stream Mapping to reduce lead-time in software product customizations and provided some hints and tips as how different steps in Value Stream Mapping (drawing current state map, identifying waste etc) can be performed.

The concept of “waste” is the foundation of the lean philosophy. Traditionally waste is defined as any activity that consumes time, resources, or space but does not add value to the product as perceived by the customer. All lean approaches focus on identifying and eliminating waste (see [223] for further details on the measurement and visualization of the flow of lean software development). However, most of the literature reporting use of Value Stream Mapping talks about the seven wastes purely in the manufacturing context. Whereas, in the context of software
development, there can be activities and steps that do not add value directly and thus may be termed as waste. For example, in manufacturing or engineering, performing detailed analysis to ensure architectural flexibility and maintainability for future sustainability might be termed as a waste. Whereas, within the software development context, it could be highly value-adding for future additions and modifications consequently reducing lead-time for future releases. Moreover, it is fundamental to realize that a reduction in lead-time might be negatively impacting other value considerations, such as quality of the product and architecture, human capital value, innovation capability to name a few. Thus, in an effort to eliminate waste, extreme caution has to be taken. To the best of our knowledge, this has not been discussed explicitly in the existing literature related to Value Stream Mapping.

3. Contribution 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 - Evaluating Evidence of Industry Application and Empirical Validation of Domain Analysis Solutions for Software Product Lines

Chapter 2 contains the systematic review of modeling and scoping activities for 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 1.5.7 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:

  - C1: Identification of appropriate domain analysis solution
Chapter 3 - Evaluating Alignment between Stakeholders

Chapter 3 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 [20-22]. 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 [20, 21]. 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 [118] (MERTS 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 homogeneously, and the same strategies need to be the basis for both the planning and the development of a product [118]. 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 3 – main contributions in comparison to the software product management challenges presented in Section 1.5.7 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.
  - C2: Creation of Product Value
  - C3: Alignment with respect to Product Strategy

Chapter 4 – A Homogenous and Consolidated View of Software Value

Chapter 4 presents a consolidated view (called the Software Value Map) of the software value concept. The Software Value Map is presented utilizing four major perspectives, (i) financial, (ii) customer, (iii) internal business process, and (iv) innovation and learning. Two of the six root causes, identified during application of MASS at Ericsson to evaluate alignment and identify root causes of misalignments, were “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 little discussions about long term strategy between SPMs” and “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”.

A product’s strategic goals should be defined based on the values important for stakeholders to ensure a win-win situation [224] (see Section 1.3.5). However, practitioners discussing value often talk about either different things or have too shallow understanding/view on the concept of value itself. This can be attributed to the lack of common definitions and interpretations of the concept of software value. This also leads to the fact that short-term gains, for example revenues or immediate customer satisfaction, are the dominant value aspects taken into consideration, while other aspects such as architectural impact or long-term customer lock-in, commissioning efficiency, and so on go
largely ignored. A common understanding of the concept of value, as well as a way in which this complex value aspects can be used in daily decision-making are central in the development of software intensive systems.

Chapter 4 – main contributions in comparison to the software product management challenges, presented in Section 1.5.7, are:

- Presents a consolidated view (called the Software Value Map) of the software value concept is presented utilizing four major perspectives, (i) financial, (ii) customer, (iii) internal business process, and (iv) innovation and learning. The value map offers a unified view of value where value concepts are categorized as value aspects, sub-aspects and value components, which can be used by professionals to develop a common understanding of value, as well as acting as decision support to assure no value perspective is unintentionally overlooked when taking product management and development decisions

- Maps links between different value aspects - making interrelationships explicit

- Shows how the proposed Software Value Map can be used to construct a number of Impact evaluation pattern, in essence a tailoring of the value map to an industry case and evaluation of the value map

  - C2: Creation of Product Value
  - C3: Alignment with respect to Product Strategy

Chapter 4 - main contribution in comparison to the aspects to be considered while value-based decision-making, presented in Section 1.3, is

  - A4: Consolidated view of value aspects relevant from different perspectives for decision-making

Chapter 5 – Value Stream Mapping Framework for Software Engineering

As global competitiveness and shorter time-to-market come to the software development industry, the search is on for a better way to create first-class software rapidly, repeatedly, and reliably. Lean initiatives in manufacturing, logistics, services, and product development have led to dramatic improvements in cost, quality and delivery time [48]; do lean practices have the potential to do the same for software development? The short answer is Absolutely! (see, for example, [218, 225]). Of the many methods that have arisen to improve software engineering processes, Lean is emerging as one that is grounded in decades of work understanding how to make processes better. Lean thinking focuses on giving customers what they want, when and where they want it, without a wasted motion or wasted minute.

One of the seven lean software development tools is Value Stream Mapping [220, 222]. Value Stream Mapping is used to achieve the one of the three goals of Lean software development, i.e. efficient engineering processes. A value stream contains all the tasks and actions (both value added and non-value added) currently required to bring a product through the main process steps to the customer a.k.a end-to-end flow of process. The lean philosophy emphasizes to optimize the whole value stream, from the time it receives an order to address a customer until the software is deployed and the need is addressed because if a company focuses on optimizing something less than the entire value stream, the overall value stream is guaranteed to suffer [47]. In the existing recommendations for application of Value Stream Mapping (see [220]), indirect value adding activities would be termed as non-value adding. In traditional manufacturing, a task adds value directly to a process or product whereas in a software development context, there can be several tasks that do not directly add to a product or process value (for example a task improving maintainability of the architecture) however; they do add value to the business by making the product more sustainable. Thus, the
recommendation from software product development perspective is to evaluate the value from all the four perspectives: customer, internal business, innovation and financial (value aspects and value components that can be evaluated within each perspective are given in the Software Value Map, see Chapter 4). Moreover, the framework proposed in this chapter explicitly lifts the importance of having both process chefs (managers) and process doers (e.g. developers, testers etc) involved in the software process improvement initiative.

Chapter 5 – main contributions in comparison to the challenges and success factors, presented in Section 1.6.4.4, are:

- Presents explicit guidelines as how the Software Value Map can be used during Value Stream Mapping to measure/evaluate the effects of suggested improvements for waste elimination. Through the use of the value map, it is ensured that the impact of improvement on organization and product also focuses on long-term value considerations in addition to the traditional short-term project focused value considerations (time and cost).
  - F3: Commitment and involvement to SPI by management
  - F4: Commitment to SPI and Involvement in the SPI initiative by process doers
  - F6: Measurement of improvement effects (considering all value aspects derived from the defined goals).

Chapter 6 – Innovative Features Selection using Real Options Theory

Chapter 6 proposes the use of real options theory to support software product managers to decide whether to make an investment in an innovative feature or not. This approach creates a richer decision space, allowing for more informed decision-making, leading to greater return on potential. In today’s competitive environment, innovations play a major role for any product in the market by creating and sustaining competitive advantage, opportunities for a company to evolve market trends, and expand the client base. However, for realizing these innovative features, market uncertainty can become a major risk. A strategic and proactive approach to decision-making becomes essential which consequently would increase a product manager’s responsibilities. He has to think and act by analyzing various possibilities and perspectives to balance product value, company vision and mission, targeting and creating a market, killing market competition, and enhancing a product’s portfolio; but all in a specified time with the available budget and resources keeping in view the with the inherent uncertainty and risks involved [42].

Chapter 6 – main contribution in comparison to the software product management challenges, presented in Section 1.5.7, is:

- Presents a valuation method that can be used to analyze different future possibilities and make decisions other than now or never, taking uncertainty explicitly into account, something often missing in current decision-making
  - C4: Selection of Innovative features

Chapter 6 – main contribution in comparison to the aspects to be considered while value-based decision-making, presented in Section 1.3, is:

- Exemplifies how use of real options can provide a richer option space that can be understood and used by all stakeholders from revenue management to product management to developers.
  - A5: Richer option space
3.1 Contribution Statement
Mrs. Mahvish Khurum is the main author for all the five papers. This includes the responsibility for running the research process, discussions with the co-authors, and conducting most of the writing of the papers. The research in Chapter 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 presented in Chapter 2 was designed by Mahvish Khurum with the help of Dr. Tony Gorschek. The study reported in Chapter 3 was designed and conducted entirely by Mahvish Khurum. While the analysis and writing was done by Mahvish Khurum assisted by Dr. Tony Gorschek. The Software Value Map presented in Chapter 4 was developed primarily by Mahvish Khurum assisted by Tony Gorschek with input from Magnus Wilson. The industrial evaluation reported in Chapter 4 was conducted by Mahvish Khurum and analysis and writing was done by Mahvish Khurum assisted by Dr. Tony Gorschek and reviewed by Magnus Wilson. The framework proposed in Chapter 5 was designed by Mahvish Khurum and Kai Petersen with Mahvish’s major contribution was in designing the use of the Software Value Map for Value Stream Mapping. Kai Petersen was the main driver of the study reported in Chapter 5 (both conducting the study and analyzing the results). The writing was done by Mahvish Khurum assisted by Dr. Kai Petersen and reviewed by Dr. Tony Gorschek. The approach proposed in Chapter 6 was designed by Mahvish Khurum with the assistance of Dr. Sebastian Barney. The writing was done mostly by Mahvish Khurum and partly by Dr. Sebastian Barney. The paper was reviewed by Dr. Sebastian Barney and Dr. Tony Gorschek.

4. Research Questions
The research mission statement posed for the whole thesis in a way summarizes all the contributions. This statement can be seen as driving all of the following efforts. It is formulated as:

**To what extent can the concept of value-based decision support help improve decision-making for software product management?**

This mission statement binds the field, but opens up for 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) and confirmed through a series of exploratory case studies in industry. The research questions, in this section, are on a macro level, while they are more specific in the respective chapters.

The product line approach is recognized as a successful value-based approach for reuse in strategic software development [226, 227] with the major benefits of product lines adoption reported as reduced time to market [227], reduced cost [28] and improved quality [28, 228]. For these reasons, many companies developing software intensive products have either adopted or are considering the adoption of a software product line approach [227, 229]. 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 selecting a particular domain analysis solution to be used in the industry, it is important for software product managers to analyze the level of industrial application and/or empirical evaluation 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 C1). This gave rise to Research Question (RQ) 1:
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 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.

In the context of market-driven product development, product strategies are the main tool for planning and realizing the goals and objectives of a product [20-22]. 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 [20, 21]. 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 [118] (MERTS is one method for formulating product strategies). However, to ensure that a product’s strategy is explicitly understood and used homogenously while making decisions related to supplying product value requires investigation. The degree of alignment between involved stakeholders with respect to the understanding and interpretation of a product’s strategy (Challenge C5) is needed to be studied and understood. This led to RQ2:

RQ2: How can the degree of alignment, among and between involved stakeholders, with respect to a product’s strategic goals and objectives, be evaluated?

In order to answer this RQ, Chapter 3 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.

A product’s strategic goals should be defined based on the values important for stakeholders to ensure a win-win situation [13]. However, practitioners discussing value often talk about either different things, or have too shallow understanding/view on the subject. This also leads to the fact that short-term gains, for example revenues or immediate customer satisfaction, are the dominant value aspects taken into consideration, while other aspects such as architectural impact or long-term customer lock-in, commissioning efficiency, and so on go largely ignored. A common understanding of the concept of value, as well as a way in which this complex value aspects can be used in daily decision-making are central in the development of software intensive systems. This gives rise to following research question:

RQ3: In the context of software product development and management, how can decision support be provided for value-based decision-making?
In order to answer this research question, there is a need to paint a complete picture of value components affecting decision-making from different perspectives: customer, internal business, innovation and learning. Such a picture can help in the visualization of long term strategic goals and objectives based on value components important for the organization.

RQ3.1: What are the different value aspects that need to be considered, utilizing different perspectives, while making decisions about software product development and management?

RQ3.2: How can different value aspects be measured and analyzed together to evaluate their effect on decision-making?

The first step towards answering the above mentioned research questions was to perform systematic mapping of existing software engineering literature to identify state-of-the-art in literature. After systematic mapping it was found that, in the existing literature and research, several researchers have presented value components and corresponding valuation/measurement solutions needed for making decisions about software product development. However, the contributions are often isolated and have a limited perspective, focusing on e.g. only cost, or only product’s characteristics like usability. Consequently, a complete picture of value components relevant from different perspectives required for taking software product management and development decisions is missing. This lead to the creation of the Software Value Map (Chapter 4) which is an exhaustive collection of value aspects and value components relevant from different perspectives for decision-making.

After creation of the Software Value Map, a natural progression towards it evaluation was to use it when taking different product management and development decisions.

RQ3.3: How can Software Value Map be used for taking different product management and development decisions?

This lead to the creation of Impact evaluation patterns for different product development decision-making scenarios (for example requirements selection, product customization analysis, reuse etc), see Chapter 4.

Whenever a software process improvement initiative (SPI) is undertaken, suggested improvements can impact (positively, negatively or no impact) different values considerations (for example lead time, architecture’s maintainability, human capital etc). Consequently, there is a need to identify value considerations affected during an SPI initiative and the positive and/or negative effects of suggested improvements on those value considerations need to be carefully evaluated and analyzed.

RQ3.4: How can Software Value Map be used to select and evaluate a set of value components affected by an SPI initiative?

This question was answered by using the value components from the Software Value Map for impact evaluation of improvement suggestions identified during Value Stream Mapping (an SPI initiative). The details are given in Chapter 5.

Innovation is one of the important value considerations for enabling product differentiation and increasing market growth. However, determining the value of an innovative feature is a difficult task due to the number of risks and uncertainties involved. In order to support software product managers to decide whether to make an investment in an innovative feature or not by providing them with a
richer decision space, allowing for more informed decision-making, leading to greater return on potential, the following sub-research question was answered in Chapter 6:

RQ3.5: How can real options theory be used to do innovative features selection?

5. Research Approach and Methodology

Research methodology provides an outlook and concrete methods/models/practices in relation to extracting results/data used to answer the research questions [230]. Thus, a suitable methodology based on the nature of reality and the nature of knowledge (according to the research paradigm, see Section 5.1) must be chosen to collect the necessary data for answering the research questions.

Research paradigms are composed of “assumptions about knowledge and how to acquire it and about the physical and social world” [231, 232]. Guba and Lincoln [233] propose three questions that need to be addressed in defining a paradigm: What is the nature of the reality that is addressed (ontology)?; what is the nature of the knowledge (epistemology)?; and what is the best approach to obtaining the desired knowledge and understanding (methodology)?.

A brief introduction to the research paradigms and underlying philosophies is given below. This is deemed as necessary to motivate the type of research questions posed and research methods used in this thesis to answer the research questions.

5.1 Research Paradigms

5.1.1 Positivist

The Positivist/Postpositivist paradigm follows the empiricist approach, with the assumption that social worlds are comparable to the natural world, and as such can be studied using similar principles. Reichardt and Rallis [234] suggest that, under this paradigm, explanations of a causal nature can be suggested by grounding these explanations in theory. Methodologies such as experiments, quasi-experiments, correlational and causal comparative studies [235] support this research paradigm.

The ideas associated with positivism have been developed and challenged, stated, re-examined and re-stated over time. Outhwaite [236] suggest that there are three distinct generations of positivist philosophy:

1. First generation produced philosophers such as Locke, Hume and Comte [237, 238] who were associated with the early traditions of positivism that were established in 18th and 19th century. Basic assumption of positivism is that an objective reality exists which is independent of human behavior and is, therefore, not a creation of the human mind. The senses are used to accumulate data that are objective, evident and measurable, anything else should be rejected as transcendental [237].

2. Second generation called logical positivism is associated with philosophers of the early 20th century collectively known as the Vienna Circle [239, 240]. Logical positivist stressed the importance of induction and verification and the establishment of laws to cleanse scientific knowledge of speculative and subjective viewpoints by stress. It attempts to achieve this by the use of mathematics and formal logic to provide analytical statements about the observed world.

3. Third generation is commonly associated with Karl Hemple [241], developed in post-war period. The third generation of positivists focused on the need for reasoning that moves from theoretical ideas or a set of given premises, to a logical conclusion through deductive thinking.
The general elements of positivist philosophy have a number of implications for social research which are given below (adapted from Easterby-Smith et al. [242] and Hughes [243]):

- **Methodological**: all research should be quantitative, and only research which is quantitative can be the basis for valid generalizations and laws
- **Value-freedom**: the choice of what to study, and how to study should be determined by objective criteria rather than human beliefs and interests
- **Causality**: the aim should be to identify causal explanations and fundamental laws that can explain human behavior
- **Operationalization**: concepts need to be operationalized in a way that enables facts to be measured quantitatively
- **Independence**: the role of the researcher is independent of the subject under examination
- **Reductionism**: problems are better understood if they are reduced to the simplest possible elements

In summary positivism is based upon values of reason, truth and validity and there is a focus on the facts purely, gathered through surveys and experiments and analyzed statistically [242, 244, 245]. The exploration and examination of human behavior, such as feelings and biases, are beyond the scope of positivism.

### 5.1.2 Interpretivist/Constructivist

Popper [246] questioned the positivist claims to truth and scientific knowledge through the process of induction. According to Popper, falsification, that is, the disproving of theories and laws, was much more useful than verification, as it provided more purposeful research questions and practices [246].

A major criticism of the positivist approach is that it does not provide the means to examine human beings and their behavior in-depth. In the social world, individuals and groups make sense of situations based on their individual experiences, memories and expectations [244, 247]. Therefore, meaning is constructed, and over time constantly reconstructed, through experiences resulting in many different interpretations. Thus, it is fundamental to understand these meanings and the contextual factors that influence, determine and effect the interpretations made by different individuals. Interpretivists consider that there are multiple realities since ‘all knowledge is relative to the knower’ [247]. This paradigm emerged from early efforts in phenomenology and interpretive understanding, presently also referred to as hermeneutics [248]. Tesch identified twenty-six types of methods associated with this paradigm, including ethnographic methods, hermeneutic methods, phenomenological methods and naturalistic and case studies methods [249].

Within the interpretivist/constructivist paradigm, the focus of the researcher is on understanding the meaning and interpretations of social actors and to understand the world from their viewpoint. This is highly contextual and, therefore, not generalizable [245]. This paradigm is associated with qualitative approaches to data gathering [250].

Table 1 gives a summary of the contrasting beliefs associated with major research paradigms.
Table 1 - Contrasting beliefs associated with major research paradigms

<table>
<thead>
<tr>
<th>Beliefs</th>
<th>Postivist/postpositivist</th>
<th>Interpretive/Constructivist</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ontology</strong> - What is the nature of reality?</td>
<td>One reality; observable through probability evaluations</td>
<td>Multiple socially constructed realities</td>
</tr>
<tr>
<td><strong>Epistemology</strong> – What is the nature of knowledge?</td>
<td>Objectivity is important; researcher manipulates and observes in dispassionate objective manner</td>
<td>Interactive link between researcher and participants; values are made explicit; created findings</td>
</tr>
<tr>
<td><strong>Methodology</strong> – What is the approach for obtaining the desired knowledge and understanding?</td>
<td>Quantitative (primarily); interventionist; decontextualized</td>
<td>Qualitative (primarily); hemeneutical; dialectical; contextual factors are described</td>
</tr>
</tbody>
</table>

Since the contribution of this thesis is to provide value-based decision support for software product managers (a role performed by humans) empirical interpretive research was the chosen paradigm as social activities make up a significant component of the decision-making processes: requirements selection, portfolio management, strategic alignment and software process improvement.

5.2 Empirical Research Design and Research Contribution

Research design concerns the various components that should be thought about and kept in mind when planning and carrying out a research project. The components of research design are detailed below.

5.2.1 Purpose(s) of enquiry

The purpose answers the questions: “what is this study trying to achieve?”, and “why is it being done?” [251]. Purpose of enquiry is classified and detailed in Table 2.

Table 2 – Classification of purpose of inquiry [251]

<table>
<thead>
<tr>
<th>Purpose of Inquiry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exploratory</strong></td>
<td>• To find out what is happening, particularly in cases where the context and situations are not well understood</td>
</tr>
<tr>
<td></td>
<td>• To seek new insights</td>
</tr>
<tr>
<td></td>
<td>• To ask questions</td>
</tr>
<tr>
<td></td>
<td>• To assess phenomena in a new light</td>
</tr>
<tr>
<td><strong>Descriptive</strong></td>
<td>• To portray an accurate profile of persons, events or situations</td>
</tr>
<tr>
<td></td>
<td>• Requires extensive previous knowledge of the situation etc. to be researched or described. This is required to ensure that a researcher knows appropriate aspects on which to gather information</td>
</tr>
<tr>
<td><strong>Explanatory</strong></td>
<td>• Searches for an explanation of a situation or problem, traditionally but not necessarily in the form of causal relationships</td>
</tr>
<tr>
<td></td>
<td>• To explain the patterns relating to the phenomenon being researched</td>
</tr>
<tr>
<td></td>
<td>• To identify relationship between different aspects of phenomenon</td>
</tr>
<tr>
<td><strong>Emancipatory</strong></td>
<td>• To create opportunities and the will to engage in social action</td>
</tr>
</tbody>
</table>
### Purpose of Inquiry

<table>
<thead>
<tr>
<th>Purpose of Inquiry</th>
<th>Description</th>
</tr>
</thead>
</table>
| Evaluate          | • To do comparisons  
                     • To assess effectiveness or appropriateness of a solution  
                     • To improve whatever is being evaluated |

### 5.2.2 Theoretical perspective

What theory will guide or inform the study? How will the researcher understand the findings? What conceptual framework links the phenomenon being studied?

### 5.2.3 Research Questions

To what questions is the research geared to provide answers? What the researcher needs to know to achieve the purpose(s) of the study? What is it feasible to ask given the time and resources that are available to the researcher?

### 5.2.4 Research Design

Research designs are different ways to establish that the desired phenomenon is observed and influenced as intended [251, 252]. The motivation is to be able to, with as much certainty as possible, separate the effects 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 [251]. Generally, data collected in studies following fixed designs are quantitative, while the data collected in studies based on flexible designs are qualitative [251]. However, this might not always be the case, for example, data collected through case studies can be both qualitative and quantitative (see Chapters 3-5).

#### 5.2.4.1 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 [251] fixed designs can be divided into: true experiments, quasi-experiments, single case experiments and non-experimental fixed designs. Systematic reviews can also be termed as fixed design research. The overall design, if done rigidly, lends itself towards being of fixed type (like in the field of medicine), however, much of the operation of a SLR is based on expert judgment and should be recognized as potential threats. These threats can however be alleviated through e.g. iterative development or criteria and agreement measurement through e.g. kappa analysis.

A literature review is usually the first step in any research and development activity [253]. Through a literature review, a researcher can map existing and previously developed knowledge and initiatives in a particular research field [253]. Due to the growth of scientific production, the role of literature reviews has proportionally grown [253], thus general rules for performing a literature review have evolved. A systematic review is a specific scientific methodology that goes one step further than the simple literature overviews. A systematic review can be defined as a means of evaluating and interpreting 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 [191]. The guidelines for performing systematic reviews presented by Kitchenham [191] are: planning the review, conducting the review and reporting the results of the review.
Using true experiment design a treatment can be compared to a control group or to another treatment group [254]. An investigation is termed as a pure experiment if the subjects are randomly chosen and 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 [254].

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 [251]. 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.

Mostly data collected through fixed designs is quantitative, descriptive statistics (i.e. measures of central tendency, measures of variability, standard deviation error bars) and statistical methods (such as cross-tabulation, chi-square tests, correlation coefficients, analysis of covariance, multiple regression, multivariate analysis) can be employed to analyze collected data.

5.2.4.2 Flexible Designs
Flexible designs are generally used in situations where the theory is being constructed from the perception of an individual or a group [252]. These types of designs are employed in research that 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 a single case. The data in a case study can be collected in multiple ways and is mostly qualitative. The collected data can be analyzed using descriptions, themes and assertions [251]. 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 [252, 255]. In ethnographic studies people in their specific context are studied. This type of study employs observation over a longer period of time. The data is primarily collected through observations and interviews. Data is organized using coding (event coding, state coding, interval coding) [251]. Typical approaches for data analysis are descriptive and interpretive by coding the data [251]. The limitation is that the results of the observation difficult to present in an objective manner [251].

In grounded theory design the focus of the study is on the evolution of a theory, which explains what is being observed [256]. The subjects of 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 typically through interviews with 20-30 individuals to saturate categories and detail a theory [256]. Data analysis can be done through open coding, selective coding or axial coding. 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.

In general the purpose(s) and research questions together influence the choice of research design. For example,

- If the purpose of inquiry is exploratory, flexible research designs are appropriate. “What” questions concerning “what is going on here”, “how” and “why” questions normally lend themselves to some form of flexible design.
• If purpose of enquiry is to describe, non-experimental fixed designs are appropriate. “What” questions asking “how many”, “how much”, “who” and “where” suggest the use of a non-experimental fixed design for example survey.

• If the purpose of enquiry is to explain, experiments are appropriate. If the researcher has control over the events, quantitative data can be gathered and there is substantial knowledge about the likely mechanism involved, then experiments can be used to answer “how” and “why” questions [251].

• If the purpose of enquiry is to evaluate, depending on the epistemology, experiments or case studies are appropriate. If the purpose is to do comparisons, experimentation design is more suited. On the other hand if the purpose is to assess the effectiveness and appropriateness of a solution in a specific setting (i.e. it is a ‘case’ rather than a sample), case study strategy is more appropriate for such evaluations [251].

5.2.5 Data Sources
In addition to the choices to be made while selecting research design as detailed above, the researcher also needs to select the context and subjects to be studied. Glass [257] discussed the “software research crisis”, concluding that the central problem with software research is that most is not relevant to industry. Since then there has been a movement towards more applied research in the software development industry, helping to ensure that research is valuable to the people who develop software. However, recent work would suggest that progress is slow [258, 259].

At the same time, it is important to realize that it is not always possible to conduct research in an industrial context due to certain limitations. For example, it might not be possible to control the required variables to conduct an experiment in industry due to a business’ need to meet certain tight deadlines. In such cases, it is necessary to conduct the research in for example laboratory settings. It is also important to consolidate the extensive knowledge in a given research area. Systematic reviews are becoming increasingly common technique to achieve this goals within software engineering discipline [191, 260].

5.2.6 Sampling Strategy
Sampling is that part of statistical practice concerned with the selection of a subset of individuals from within a population to yield some knowledge about the whole population, especially for the purposes of making predictions based on statistical inference [251]. A sampling strategy basically answers the following questions: from whom will the researcher seek data, where and when, how does the researcher balance the need to be selective with the need to collect all the data required? [251]

The major groups of sample designs are probability sampling and non-probability sampling:

1. Probability sampling – includes some form of random selection in choosing the elements. Greater confidence can be placed in the representativeness of probability samples. This type of sampling requires a selection process in which each element in the population has an equal and independent chance of being selected. Quantitative research uses probability sampling as the aim is to produce a statistically representative sample or draw statistical inference. Four main methods include: 1) simple random, 2) stratified random, 3) cluster, and 4) systematic.

2. Non-probability sampling – the elements that make up the sample, are selected by non random methods [261]. This type of sampling is less likely than probability sampling to produce representative samples. Qualitative research uses non-probability sampling as it
does not aim to produce a statistically representative sample or draw statistical inference. The three main methods are: 1) convenience, 2) quota, and 3) purposive.

5.2.7 Research Contribution
Research papers can be categorized into one of the following categories based on their contributions [262].

- **Evaluation Research**: Techniques are implemented in practice and an evaluation of the technique is conducted, i.e., it is shown how the technique is implemented in practice (solution implementation) and what are the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). This also includes problem identification in industry.

- **Validation Research**: Techniques investigated are novel and have not yet been implemented in practice. Techniques used are for example experiments, i.e., work done in the laboratory.

- **Philosophical Research**: These papers sketch a new way of looking at existing things by structuring the field in form of a taxonomy or conceptual framework.

- **Solution Proposal**: A solution for a problem is proposed, the solution can be either novel or significant extension of an existing technique. The potential benefits and the applicability of the solution are shown by an example or a good line of argumentation.

- **Experience Research**: What and how something has been done in practice? It has to be the personal experience of the author.

- **Opinion Papers**: These papers express the personal opinion of somebody whether a certain technique is good or bad, or how things should been done. They do not rely on related work and research methodologies.

5.3 Research Setting
The primary setting used for the collection of data in relation to the results presented in this thesis is summarized below.

5.3.1 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 3-5. 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, five major software-intensive products 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.

5.4 Categorization of Chapters
The research presented in this thesis employs a mixed methodology approach.
Table 3 – Categorization of chapters with respect to research components and contribution

<table>
<thead>
<tr>
<th>Components of Research</th>
<th>Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Purpose of Inquiry</td>
<td></td>
</tr>
<tr>
<td>Exploratory</td>
<td>X</td>
</tr>
<tr>
<td>Descriptive</td>
<td></td>
</tr>
<tr>
<td>Explanatory</td>
<td></td>
</tr>
<tr>
<td>Emancipatory</td>
<td></td>
</tr>
<tr>
<td>Evaluate</td>
<td>X</td>
</tr>
<tr>
<td>Research Type</td>
<td></td>
</tr>
<tr>
<td>Qualitative</td>
<td>X</td>
</tr>
<tr>
<td>Quantitative</td>
<td>X</td>
</tr>
<tr>
<td>Research Design</td>
<td></td>
</tr>
<tr>
<td>Experiment</td>
<td></td>
</tr>
<tr>
<td>Survey</td>
<td>X</td>
</tr>
<tr>
<td>Case study</td>
<td></td>
</tr>
<tr>
<td>Ethnography</td>
<td></td>
</tr>
<tr>
<td>Grounded theory</td>
<td></td>
</tr>
<tr>
<td>Research Environment</td>
<td></td>
</tr>
<tr>
<td>Industry</td>
<td></td>
</tr>
<tr>
<td>Laboratory</td>
<td></td>
</tr>
<tr>
<td>Literature</td>
<td>X</td>
</tr>
<tr>
<td>Research contribution</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td>X</td>
</tr>
<tr>
<td>Validation research</td>
<td></td>
</tr>
<tr>
<td>Philosophical</td>
<td></td>
</tr>
<tr>
<td>Solution proposal</td>
<td></td>
</tr>
<tr>
<td>Experience papers</td>
<td></td>
</tr>
</tbody>
</table>

This is due to the fact that close collaboration with an industry partner to some extent dictates the limitations and amount of the data collected. An overview of the methodologies used in each chapter of this thesis is presented in this section; further details on the methodologies used can be
found in each chapter. A summary of the research methodologies, collected data types and context in which research was conducted is presented in Table 3.

Chapter 2 presents a systematic review of literature on domain analysis for software product lines. The purpose of this enquiry was to evaluate the strength of empirical evidence, and usability and usefulness of the existing domain analysis solutions in state-of-the-art in literature. A set of selected primary studies was classified according to a pre-defined schema, providing quantitative data on the research publications in this area. A descriptive analysis of the empirical evidence and usability and usefulness is provided. Thus it can be seen as an evaluation, even though not empirical in the sense of applying the research first hand in an application scenario.

Chapter 3 uses qualitative methods to evaluate the alignment between success critical stakeholders with respect to product strategies using the MASS method, and quantitative methods to determine the degree of alignment with respect to specific strategic goals of the products. A case study setting was used with a set of qualitative questions and closed questions. The contribution type is evaluation research.

Chapter 4 presents a consolidated view on value aspects relevant for taking software product management and development decisions. A Software Value Map was created using systematic mapping and snowball sampling in combination. The method involved conducting a survey of the literature addressing value aspects considered during software development decision-making. Value aspects were extracted from the selected studies. The resulting map was subsequently evaluated through the creation of impact evaluation patterns in an industrial case study involving a series of work sessions. The work sessions included a set of semi-structured questions to identify current decision-making criteria and a small survey questionnaire to identify value aspects most relevant for the participants’ decision-making. Qualitative data collected was analyzed and descriptive analysis of current decision-making criteria is presented followed by pattern creation through quantitative analysis of the questionnaire results. The contribution type is evaluation research.

Chapter 5 presents a Value Stream Mapping Framework for the Software Engineering context. Usability and usefulness of the proposed framework was evaluated in an industrial case study. The results of the case study and lessons learned from applying the proposed framework were drawn out to provide the descriptive analysis. The contribution type is evaluation research.

Chapter 6 presents an example to explore the use of real options theory for the selection of innovative features. A solution proposal was presented for using real options for innovative features selection. The contribution type is a solution proposal.

5.5 Research Approach

The research presented in this thesis has been performed in close collaboration with industry. In this context, it is necessary to evaluate the current industrial work practices (in the thesis context decision-making practices) before suggesting any improvements. Therefore, a specific “macro” technology and knowledge transfer research model presented by Gorschek et al. [57] (see Figure 6) was used to enable industry-based research and solution development.

Looking at the technology and knowledge transfer model (see Figure 6), there are a number of steps. Step 1 is to identify the problem/issue in an industry which can be done using different methods e.g. exploratory case studies, light weight 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 [57].
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 the 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).

Chapter 2 is a systematic review of the current literature on domain analysis solutions for software product lines. The research presented in Chapters 3-5 follows the technology transfer model presented by Gorschek et al. [57].

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 [57]. 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 6).

The need for a consolidated view on value was identified based on the root causes for misalignments between stakeholders (see Section 3). The Software Value Map was created after studying state-of-the-art literature (Step 2). The map is a collection of value aspects and value components that need to be evaluated during practical decision-making; therefore, it was difficult to simulate the decision-making scenarios faced by product managers in a laboratory environment as it demands relevant experience and expertise. Thus, the Software Value Map has been evaluated using case study research method at Ericsson (Step 5).

The need to incorporate evaluation of value aspects other than lead-time and cost during value stream mapping was also identified during exploratory interviews at Ericsson (Step 1). The state-of-the-art
literature was read (Step 2) and the traditional Value Stream Mapping was modified to fit into software engineering context (Step 3). The proposed Value Stream Mapping framework was evaluated in real industrial context (Step 6) since it was difficult to simulate process assessment and improvement initiatives in a lab environment.

Chapter 6 contains a solution proposal for selecting innovative requirements using real options theory. The basis for the solution stemmed from the challenge faced by product managers in industry when selecting innovative requirements (identified in Step 1) and the proposed solution (Step 3) is based on the state-of-the-art literature (Step 2); validation and release of the proposed solution (Steps 4-7) remains a future work.

6. Conclusions

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

One of the central conclusions from the research presented in this thesis is that it is challenging to apply value-based approach in decision-making. The primary reasons being that value in itself is a multi-dimensional concept, often vaguely defined, and the stakeholders involved in decision-making might have different perspectives and views on different value aspects. This was confirmed through literature review and empirical case studies (see Chapter 3 and Chapter 4). A value for one might be trash for someone else. In order to address this challenge, the Software Value Map was created. This map contains a consolidated view of all value aspects that can be relevant for decision-making for software development and management. Based on the map, impact evaluation patterns can be created to acknowledge different perspectives and evaluations by stakeholders on value aspects relevant for decision-making. As a step towards making the Software Value Map usable, patterns, critical paths, and themes were introduced extending the map.

According to initial evaluations, the Software Value Map, impact evaluation patterns, critical paths and themes, seem to be very promising in offering decision support for value-based decision-making. However, extensive pilot tests are required to further evaluate usability and usefulness. Furthermore, the Software Value Map has been used, as a basis for evaluation of process improvement initiatives, to identify potential value aspects that should be evaluated, in addition to reduction in lead-times and costs. Practitioners perceived the use of the Software Value Map, for an overall evaluation of all the relevant value aspects, as extremely useful. They realized, during the value stream mapping activity, that in addition to cost and lead-times, evaluation of other value aspects, enables evaluation of process improvement suggestions in a wider context.

A potential weakness, in the proposed evaluation technique (rubrics-based) to evaluate the impact on value aspects, is that the evaluation is still based on expert judgment, and therefore, subjective. To gauge and compare the evaluations made by practitioners, objective evaluations, or at least objective measures complementing expert judgment could be beneficial. However, the introduction of more objective measures and evaluations is extremely complex as there are many tangible but also intangible value aspects. This is further complicated by the need for different scales and units in which value aspects are measured (for example, revenues in dollars and understandability value of the product as the number of features understood over the total number of features in the product). Thus, the challenges encompass; 1) finding measures and metrics that can quantify intangible value aspects, and 2) common scales and units of measurements for comparing different value aspects. Addressing these challenges and using objective evaluations together with expert judgment is a way forward.

An important contribution of this thesis is that all results, whether they are in the form of systematic reviews, or the creation of methods, such as MASS or the Software Value Map, can be used by both researchers and practitioners. The detailed results of the systematic review (Chapter 2) 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 and good examples presented can inspire. 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.

The MASS method, Software Value Map and Value Stream Mapping framework are designed to operate in an industry environment. For this reason, tailorable and modularity were premiered. For example, MASS can be used by large, medium or small organizations alike to assess the degree of alignment by involving as many or as few success critical stakeholders as required. Similarly, any organization can deduce impact evaluation patterns from the Software Value Map, relevant for their decision-making scenarios. Further, the impact evaluation patterns can be stored in existing requirements engineering tools or web portal and even in simple tools, for example Excel. Moreover, the Value Stream Mapping framework for software engineering can be instantiated depending on the needs of an organization versus time and resources available. Strategies that can be used to perform different steps of Value Stream Mapping have been extensively discussed with respect to accuracy, time and resources required and complexity of the problem in hand. Recognition of an important fact that one-size-does-not-fit-all, and that one solution does not work in isolation but has to be adaptable to the overall environment, is fundamental.

However, presenting industry with research results, even logically mapped to challenges, is not enough. Prior to getting support for dynamic validation through pilots, support for technology and knowledge transfer has to be secured. This involves activities such as tool support, training, and the establishment of measurement programs to measure the effect of the introduction of a new way of working (e.g. the introduction of the Software Value Map). This has been achieved to a certain degree as much of the results presented in this thesis have been statically validated; however, dynamic validation is a logical next step. An important part in getting commitment for this depends on demonstrating the relative value of a new solution over the practices already in place. To demonstrate the usability and usefulness of the Software Value Map, the work sessions for impact evaluation patterns creation were used readily. The practitioners vividly saw that use of the Software Value Map could help them significantly in improving their decision-making, by incorporating value evaluations from different perspectives. These work sessions also played a central role in generating a great deal of interest around value-based decision-making. Similarly, the application of the Value Stream Mapping framework, proposed for software engineering was intended to map and show how to improve an end-to-end process. The mapping made it evident to the practitioners that Value Stream Mapping had great potential with respect to usability and usefulness.

These evaluations of the methods/frameworks were used by Ericsson as an input to risk assessment, which is a pre-requisite for incorporating any new practice or method in industry. Even more importantly, the evaluations were used by the researchers as an input for refining the methods/frameworks themselves, to maximize their potential for being usable and useful in a real industry setting.

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 evaluate research results and findings addressing these challenges in industry is an opportunity that should be exploited. This has been recognized by many contributors to the field over the last decade, resulting in promising and definitely relevant results, many of them used as an inspiration to the research presented in this thesis.

6.1 Validity Evaluation
All the chapters in the thesis face their own set of validity threats (see each chapter for detailed validity threats). The four perspectives of validity and threats as presented in Wohlin et. al. [254] are considered.
6.1.1 Construct validity
The construct validity is concerned with the relation between theories behind the research and the observations. The variables in the research presented in this thesis are measured through interviews, including open-ended questions where the participants were asked to express their own opinions.

The potential problem of evaluation apprehension [254] (in Chapters 3-5) was alleviated by the guarantee of anonymity as to all information divulged during the interviews, and the answers were only to be used by the researcher, i.e. not to be showed or used by any other participants, companies or researcher. Another validity threat lies in the question that asked interviewees to rank and include additional factors if the list provided to them was inadequate. Interviewees may have thought that it was easier to rank the provided factors than propose new factors, i.e. some interdependency types may be missing. The quantitative data given by respondents is subjective since it is not based on any objective measurements; there might be differences in how the questions were interpreted.

6.1.2 Conclusion validity
Threats to conclusion validity arise from the ability to draw accurate conclusions. The interviews for case studies presented in Chapters 3 and 4 were conducted with one interviewee and each interview was done in one work session. Thus, answers were not influenced by internal discussions. To ensure that the interview instrument, including the posed questions, is of high quality to obtain highly reliable measures, several pilot studies were conducted, to avoid poor questions and poor layout, prior to conducting the interviews.

6.1.3 Internal validity
These threats are related to issues that may affect the causal relationship between treatment and outcome. Threats to internal validity include instrumentation, maturation and selection threats. The potential problem of instrumentation threats (in studies presented in Chapters 3-5) was alleviated by developing the research instrument with close reference to literature. Threat to selection bias is always present when study subjects are not fully randomly sampled. However, given that interviewees were selected based on their roles by a “gate-keeper” at the Ericsson, this threat has limited effect.

6.1.4 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) 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. Thus, it is concluded that in general 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 3-5, external validity threat arises due to the fact that interpretive research was the chosen paradigm and relatively few cases are studied in industry: in case of MASS only three products in one organization (see Chapter 3), in case of the evaluation of Software Value Map and creation of impact evaluation patterns only one product in one organization and in case of Value Stream Mapping framework application only one systems of system in one organization. In order to deal with this threat, one-size-fit-all philosophy was not followed while designing MASS, impact evaluation patterns and Value Stream Mapping. The evaluation framework used in MASS can be
tailored, based on different product strategies used in different organization; the value components to be included in an impact evaluation pattern can be identified based on a given decision-making scenario and strategies to perform value stream mapping can be chosen based on accuracy required, complexity of the problem in hand and time and resources available.

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 solutions are designed on actual needs identified in industry the chances for application in more than one organization increase.

6.2 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, there arises a need for future research as how the level of empirical application and industrial validation for the solutions presented for domain analysis 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.

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 potential 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.

MASS method and the software process improvement framework (Value Stream Mapping) based on lean philosophy need to be refined. Assessment and planning are prerequisites for an SPI effort not using a prescriptive approach. Chapter 3 and Chapter 5 in this thesis address these issues by introduction and initial evaluation of assessment methods. However, there is a need to continue this effort by replicating the studies and through this improving the methods/frameworks presented.

The Software Value Map provides a consolidated view of software value concept utilizing four major perspectives, (i) financial, (ii) customer, (iii) internal business process, and (iv) innovation and learning. However, it needs to be evaluated by creating impact evaluation patterns for different decision-making scenarios in different organizations. The concept of Software Value Map appears promising, however, the current state of Software Value Map and impact evaluation pattern has barely begun to investigate the area and gauge its potential. An important aspect in relation to impact evaluation patterns is measurement of value components. To begin with, a subjective 5-point rubric scale has been proposed. However, to make the evaluations more objective and concrete there is a need to find appropriate scales and units of measurement for comparisons between different value components (which are tangible and intangible in nature). Extreme caution needs to be taken while proposing/adapting extensive measurement programs and gathering of metrics keeping in view the time and resource constraints of an organization.

Moreover, Value Stream Mapping framework needs to be instantiated in different domains and in different context. Last, but not the least, the use of real options for selecting innovative requirements needs to be evaluated in industry.

Another important aspect that needs to be investigated is the effect of biases in decision-making. During decision-making, a decision maker wants to identify as many of the alternatives as possible and choose the one that is the best fit for his/her goals, desires, values and so on. The capability of selecting the best choice during decision-making is effected by the decision makers’ perceptions of the problem and his/her choices. The perception of the problem and choices is influenced by different biases that the decision maker either has from the start or is subjected to during the decision
process [263]. Most of the biases are of cognitive nature based on human thinking, and are referred to as cognitive biases [264]. These biases range from group level behaviors to individual level. There are also emotional biases that can effect decision on emotional factors, experimenter’s biases that are based on expected results and statistical biases that are caused by errors in the choices. These biases can seriously alter a product manager’s perceptions towards value and risks. However, the effects of these biases during decision-making within software product management context have been nearly unexplored.

References


W. A. Shewhart and W. E. Deming, Statistical method from the viewpoint of quality control, University of Washington (dept), 1939.


M. Rother and J. Shook, *Learning to see: value stream mapping to create value and eliminate muda*, Lean Enterprise Institute, 2003.


Chapter 2 - Evaluating Evidence of Industry Application and Empirical Validation of Domain Analysis Solutions for Software Product Lines

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 as 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 lines have received significant attention from the software engineering community since the 1990s (Clements and Northrop, 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 Northrop, 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, Clements and Northrop, 2001, 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 Northrop, 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 Northrop, 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 Northrop, 2001). Domain analysis helps in the identification of the specification of the systems in the product line. 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 interdependencies.

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 runtime 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 Northrop, 2001). The result of a scoping activity is a scope definition document which becomes a product line core asset (Clements and Northrop, 2001). 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 Northrop, 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>
<thead>
<tr>
<th>Research Questions</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1. Are solutions, proposed for domain analysis, based on needs identified in industry?</td>
<td>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).</td>
</tr>
<tr>
<td>RQ2. Are solutions, proposed for domain analysis, applied and/or validated in a laboratory setting or in industry?</td>
<td>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).</td>
</tr>
<tr>
<td>RQ3. Are the solutions, proposed for domain analysis, usable?</td>
<td>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).</td>
</tr>
<tr>
<td>RQ4. Are the solutions, proposed for domain analysis, useful?</td>
<td>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).</td>
</tr>
</tbody>
</table>

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 (efficiency)
- Scalability of use (efficiency)

**Usefulness** is defined in terms of:

- Better alternative investment (BAI)
- Effectiveness
- 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:

- Inspec and Compendex via Engineering Village2
- ACM
- IEEEXplore
- ISI Web of Science

In order to ensure that search strings are comprehensive and precise, an expert librarian was consulted. All the 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

**Phase 1:** 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 Step 1 (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.

**Figure 3 – Steps of Phase 2 of the search strategy**

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:** The reference lists of the studies selected in Phase 2 were scanned to ensure that no relevant studies are missed. Furthermore, 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.

### 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 brainstorming 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.

<table>
<thead>
<tr>
<th>Research Type</th>
<th>Is it a new solution for domain analysis?</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Solution</td>
<td>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.</td>
</tr>
<tr>
<td>Experience report</td>
<td>Is the paper an experience report describing the introduction of product lines in a company?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Empirical Basis</th>
<th>If it is “New Solution”, is it developed based on empirically identified industry needs?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empirical</td>
<td>It is a new research idea.</td>
</tr>
<tr>
<td>Non Empirical</td>
<td>It is a new research idea.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Basis Reported as</th>
<th>If a study has empirical basis then the empirical basis reported can be categorized into</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Statements only: the authors have written statements claiming that the need for the proposed solution has been identified in industry.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application/Validation</th>
<th>If a study contains empirical application/validation, what method of application/validation was used?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empirical</td>
<td>It is applied/validated in laboratory setting or industry.</td>
</tr>
<tr>
<td>Non Empirical</td>
<td>It is not applied/validated in laboratory setting or industry.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application/Validation Method</th>
<th>If a study contains empirical application/validation, the level of explanation of the design/execution of the application/validation method used is categorized into:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. Statements: authors stated that they have applied/validated the solution in industry but no summary/details as how this was done.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application/Validation Design Explained</th>
<th>If a study contains empirical application/validation, the level of explanation of the application/validation results is categorized into:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. Nothing: no information is stated as to the results of the application/validation in the paper.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>4. Quantitative results: collected metrics and measurements are presented.</td>
</tr>
<tr>
<td></td>
<td>5. Qual + Quant: when a combination of qualitative and quantitative results are presented.</td>
</tr>
<tr>
<td>Driver of Validation</td>
<td>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.</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Replication Study</td>
<td>Is it a replication study? The answer could be Yes, No, Not clear, N/A (not applicable).</td>
</tr>
<tr>
<td>Builds on Paper(s)</td>
<td>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.</td>
</tr>
<tr>
<td>Usability &amp; Usefulness</td>
<td></td>
</tr>
<tr>
<td>Usable</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Usability Reported as</td>
<td>If a study reports usability of proposed solution, the level of usability reported can be categorized into</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>Useful</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Usefulness Reported as</td>
<td>If a study reports usefulness of proposed solution, the level of usefulness reported can be categorized into</td>
</tr>
<tr>
<td></td>
<td>1. Statements: the authors have written statements claiming usefulness e.g. “It is an effective product line validation model” (Mannion and Camara, 2003).</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>4. Qual+Quant: qualitative and quantitative data proving effectiveness of proposed solution.</td>
</tr>
<tr>
<td>Future Work Mentioned</td>
<td>Has the study made promises of future work in relation to the current research? The answer can be yes/no</td>
</tr>
<tr>
<td>Written by</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 the two researchers avoiding the potential bias and error source of having only one researcher performing the categorization.

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

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 brainstorming 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). The extracted raw data is present in Appendix C. The data labels for Figure 4 – Figure 10 have the format: data extraction category, percentage of studies from which the corresponding data was extracted, number of studies from which the corresponding data was extracted. For example in Figure 4 one of the data labels is (participation knowledge, 50%, 24) which means that 50% of the studies, claiming empirical basis for the need identified, reported “participation knowledge” as the “Basis”. The actual number of studies is 24.

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 Figure 4 which shows that a majority of the studies have mentioned identified needs as “Statements only” (42% studies), or as “Participation knowledge” (50% studies). Only 2% studies have mentioned interviewing experts to identify needs, and only 6% studies 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.

Figure 4 - Included studies, “Basis” categorization.

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. 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 5) reveals that for the years 1998 to 2007, 33% studies have used case study as an application/validation method. In 24% of the studies industry use was only stated and 36% studies 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 5, it is evident that only 5% studies have used workshops, pilots and prototyping. It is also interesting to note that for the years 1998 to 2007 only 2% studies have used experimentation as a validation method.

![Figure 5 – Number of studies categorized according to the application/validation method](image)

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 qualititative 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).

Moving on from the analysis of application/validation methods used to the analysis of the application/validation design details, Figure 6 shows the categorization of the application/validation design explanation given in the included studies claiming some form of application/validation. From Figure 6, it is possible to see that 84% studies either provide application/validation summary (50% studies) or explain application/validation in detail (34% studies). 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 Figure 7). Only 7% studies provide qualitative results as experts’ opinion, 9% studies provide quantitative results and only 3% studies provide both qualitative and quantitative results.

Thus, Figure 7 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 8 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 8 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.

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 Figure 9 it is possible to see that 80% of the studies 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 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 gave quantitative evidence of usability and only 2% of the studies 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).

Figure 9 - “Usability” reported in included studies.

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.

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 Figure 10 shows that 81% of the studies 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 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 Figure 7), 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 as case studies through workshops, interviews, or surveys. Dynamic validation (e.g. pilots) may be performed collecting metrics and qualitative data through 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


A. Jedlitschka and D. Pfahl, 2005. Reporting guidelines for controlled experiments in software engineering. Institute of Electrical and Electronics Engineers Computer Society


### Appendix A – Search Strings

<table>
<thead>
<tr>
<th>Search Strings</th>
<th>Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>( ((((&quot;product famil** OR &quot;product line&quot;) 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 &quot;asset scop&quot;) AND ((((&quot;product famil** OR &quot;product line&quot;) 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 &quot;asset scop&quot;) AND (empiric* OR experience* OR &quot;lesson learned&quot; OR &quot;lesson learnt&quot; OR &quot;lessons learned&quot; OR evaluat* OR validat* OR experiment* OR stud* OR case* OR example* OR survey OR analy*) AND (English) AND (1969-2007 WN YR) AND (English) WN LA)</td>
<td>Inspect and Compendex via Engineering Village2</td>
</tr>
<tr>
<td>(((Abstract:product and Abstract:line) OR (Abstract:product and Abstract:famil*)) AND (((requirements model*)&lt;in&gt;ab) &lt;or&gt; ((requirements engineer*)&lt;in&gt;ab) &lt;or&gt; ((requirements)&lt;in&gt;ab) &lt;or&gt; ((conceptual model*)&lt;in&gt;ab)) &lt;and&gt; (((requirements engineering)&lt;in&gt;ab) &lt;or&gt; ((requirements)&lt;in&gt;ab) &lt;or&gt; ((requirements)&lt;in&gt;ab)%20AND%20((conceptual%20model*)&lt;in&gt;ab)) &lt;and&gt; ((variability%20analysis)&lt;in&gt;ab) &lt;or&gt; ((domain%20eval*)&lt;in&gt;ab) &lt;or&gt; ((asset%20soc*)&lt;in&gt;ab)) &lt;and&gt; ((empiric*)&lt;in&gt;ab) &lt;or&gt; ((lessons%20learn*)&lt;in&gt;ab) &lt;or&gt; ((evaluat*)&lt;in&gt;ab) &lt;or&gt; ((validat*)&lt;in&gt;ab) &lt;or&gt; ((experien%)&lt;in&gt;ab) &lt;or&gt; ((case%20stud*)&lt;in&gt;ab) &lt;or&gt; ((survey*)&lt;in&gt;ab) &lt;or&gt; ((analy*)&lt;in&gt;ab) &lt;or&gt; ((investigat*)&lt;in&gt;ab) &lt;or&gt; ((demonstrat*)&lt;in&gt;ab))</td>
<td>ACM</td>
</tr>
<tr>
<td>TS=&quot;(product line&quot; OR &quot;product famil** AND (TS=&quot;(requirements&quot; OR &quot;requirements engineering&quot; OR &quot;requirements model&quot;) OR &quot;feature model&quot; OR &quot;commonality analysis&quot;) OR &quot;variability analysis&quot; OR &quot;domain scop&quot; OR &quot;domain eval&quot; OR &quot;asset scop&quot;) AND (TS=&quot;(case stud&quot; OR &quot;empiric&quot; OR &quot;experien&quot; OR &quot;Lessons learn&quot; OR &quot;evaluat&quot; or &quot;validat&quot; OR &quot;experiment&quot; OR &quot;exampl&quot; OR &quot;survey&quot; OR &quot;Analy&quot; OR &quot;investigat&quot; OR &quot;validat&quot; OR &quot;industri&quot;)) AND Language=(English) AND Document Type=(Article)</td>
<td>ISI Web of Science</td>
</tr>
</tbody>
</table>
### Appendix B – Selected Studies

<table>
<thead>
<tr>
<th>Study Id</th>
<th>Study Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>Reference</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
</tr>
<tr>
<td>Page</td>
<td>Reference</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
</tr>
<tr>
<td>Page</td>
<td>Reference</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
</tr>
<tr>
<td>Page</td>
<td>Reference</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
</tr>
</tbody>
</table>

1, 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, 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
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
Chapter 3 - Evaluating Alignment between Stakeholders

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

Tony Gorschek
Blekinge Institute of Technology
Department of Systems and Software Engineering
S-372 25, Ronneby, Sweden
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 which is very important from value-based decision making perspective even at the software development level [13].

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

Strategic alignment has many pseudonyms. It is also termed fit integration [14], [15], bridge [16], harmony [17], fusion [18] and linkage [19]. However, in every case, it is about the integration of business related strategies with their information technology (IT)/information systems (IS) strategies [20]. Assessing, achieving and maintaining strategic alignment in this context have been discussed in a number of studies [20-25]. Table 1 compares MASS with other methods/frameworks by identifying the aims and objectives of each, the focus area (information systems 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], assuring alignment, i.e. a common understanding of the company’s goals and objectives for a particular product. It is important to state that MASS does not propose antecedents of market-orientation as suggested by Ajay and Jaworski [26] rather it is a method to assess, achieve and maintain strategic alignment within a market-driven software product development organization. To the best of our knowledge no other method has been presented with this objective and context.
### Table 1 – Comparisons of Alignment Evaluation Methods/Frameworks

<table>
<thead>
<tr>
<th>Study</th>
<th>Aim/Objective</th>
<th>Focus</th>
<th>Type of strategy</th>
<th>Perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>[20]</td>
<td>Determine alignment levels by means of SA practice</td>
<td>IS</td>
<td>Business</td>
<td>Projects</td>
</tr>
<tr>
<td>[21]</td>
<td>Measure existing use of IT in organizations</td>
<td>IS</td>
<td>Business</td>
<td>Business and IT</td>
</tr>
<tr>
<td>[22]</td>
<td>Measure alignment for small firms and investigate factors that influence alignment</td>
<td>IS</td>
<td>Business</td>
<td>Business and IT</td>
</tr>
<tr>
<td>[23]</td>
<td>Identify recommendations for improving alignment based on the organization's maturity</td>
<td>IS</td>
<td>Business</td>
<td>Business and IT</td>
</tr>
<tr>
<td>[24]</td>
<td>Identify specific recommendations for improving alignment</td>
<td>IS</td>
<td>Business</td>
<td>Business and IT</td>
</tr>
<tr>
<td>MASS</td>
<td>Determine degree of alignment between SCS for creating product value aligned with a product’s strategy</td>
<td>Software product</td>
<td>Product Strategy</td>
<td>Software product</td>
</tr>
</tbody>
</table>

### 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.” [27].

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 [27] or goals [28] and short term objectives [27]. The basic aim of the goals is to set the general directions of movement, whereas objectives state the specific measures of accomplishment [28]. 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 [29] 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 [29]. 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 [30]. 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 [29]. However, as Krishnan and Karl [31] point out, if the goal is to increase the market growth/size, the targeted customers will have to 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 [29, 32]. For determining a product’s position in the market it has to be differentiated based on cost, price or value of product offering with a compromise on the remaining two. 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 [29] 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 [29]. 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 [27]. 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 [30]. He points out that a roadmap is a relatively common
way of representing targets based on development in the context of time and releases [30].

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 can be 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 product 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 MERTS
MERTS is centered on ensuring that the five strategic questions for a product are answered explicitly [7]. 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 Figure 1).

2.2.1 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 [7]. 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.

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 [7]. Priorities can be assigned to each of these factors showing their relative importance with respect to each other.

![Figure 1 - MERTS Steps](image)

**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.
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.

2.2.2 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 [30] 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 [33] can be used to support the efforts. An alternative could be to perform estimates based on previous development efforts. Additionally, requirements prioritization techniques [34] 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.
2.3 VMOST Analysis
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 [20]. It helps in
understanding how a product’s vision, mission, goals, strategies, objectives, and
tactics relate to, align with, and provide support for each other by an analyst’s
response to a number of key questions (see Section 3.1).

2.4 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>
<thead>
<tr>
<th>Objectives</th>
<th>Research Questions (and MASS evaluation questions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investigate if the SCSs understand and agree on a product’s strategic goals and objectives.</td>
<td>RQ1: How do the SCSs understand and interpret different aspects of a product’s strategy?</td>
</tr>
<tr>
<td>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:</td>
<td>RQ2: What is the degree of alignment between and among the two groups with respect to the product strategy?</td>
</tr>
<tr>
<td>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.</td>
<td>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?</td>
</tr>
<tr>
<td>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.</td>
<td>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?</td>
</tr>
<tr>
<td>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.</td>
<td>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?</td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>RQ2.5: To what degree are the groups aligned with respect to ideal priorities of a product’s strategic goals and objectives?</td>
<td></td>
</tr>
<tr>
<td>RQ2.6: What degree are the SCSs within a group aligned with respect to ideal priorities of a product’s strategic goals and objectives?</td>
<td></td>
</tr>
</tbody>
</table>
3. MASS

Figure 2 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 2 – MASS Steps](image)

3.1 Step 1 – Design Evaluation Framework
The first step is to design the alignment evaluation framework making it possible to seek answers to RQ1(*How do the SCSs understand and interpret different aspects of a product’s strategy?*) and RQ2 (*What is the degree of alignment between and among the two groups with respect to the product strategy?*). This is demonstrated in detail in Figure 3, which shows the evaluation framework used in MASS and includes both qualitative interviews and a quantitative questionnaire. RQ1 is sought through qualitative interviews utilizing the VMOST [35] technique, MERTS method and roadmapping literature.
MERTS suggests that good product strategies should contain answers to the questions stated in Section 2.1 and 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, 36]. 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 3) used in MASS evaluation can be seen in Table 3, where the combination of VMOST and MERTS is evident.

Figure 3 – Seeking answers to RQ1 and RQ2
<table>
<thead>
<tr>
<th>Method/Technique</th>
<th>Corresponding Questions</th>
<th>To Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VMOST</strong></td>
<td>1. What is the ideal end-state towards which the organization strives through the product in question (vision)?</td>
<td>Where</td>
</tr>
<tr>
<td></td>
<td>2. What is the primary activity that the organization performs to achieve the end-state?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. How are the responses to questions 1 and 2 (vision and mission) appropriate and relevant to the environment (of industry and market)?</td>
<td>How</td>
</tr>
<tr>
<td></td>
<td>4. What are the basic activities and their rationale by which organization competes with industry rivals?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. Are the responses to questions 1 and 2 (vision and mission) explicit or implicit? How?</td>
<td>Where</td>
</tr>
<tr>
<td></td>
<td>6. What goals does the organization set to determine if it is competing successfully?</td>
<td>How</td>
</tr>
<tr>
<td></td>
<td>7. What activities does the organization perform to achieve the goals in 6?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8. How do the goals in 6 support the responses to 1?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9. What are the measurable objectives that indicate achievement of goals identified in and what activities does the organization perform to achieve those objectives?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10. How do the objectives identified in 9 support the goals identified in 6?</td>
<td></td>
</tr>
<tr>
<td><strong>MERTS</strong></td>
<td>11. Is technology innovation taken into consideration for a product strategy?</td>
<td>How and What</td>
</tr>
<tr>
<td></td>
<td>12. How is technology innovation taken into consideration for a product strategy?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13. Is the existing architecture taken into consideration for a product strategy?</td>
<td>What</td>
</tr>
<tr>
<td></td>
<td>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?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15. What types of roadmaps are created for a product?</td>
<td>When</td>
</tr>
<tr>
<td></td>
<td>16. Do you have technology roadmaps in some format?</td>
<td></td>
</tr>
</tbody>
</table>

In order to cover the second main part (i.e. Part 2 in Figure 3), 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.

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). The purpose is to elicit the perception of the participants about the importance of each goal.

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.

<table>
<thead>
<tr>
<th>Value dimension</th>
<th>Usage Today</th>
<th>Today</th>
<th>Points</th>
<th>Ideal</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Goal 1</td>
<td>0 = not used, 1 = used occasionally, 2 = almost always used, 3 = is a must.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Goal 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Goal 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Goal 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Goal 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Point remaining: | 100 | Point remaining: | 100 |

If any SCS adds a new goal and other stakeholders do not add, it will result in weaker correlation between the stakeholders. However, addition of any new goal will not change consecutive MASS steps.
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 4). 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.

MASS is focused on studying the alignment between the SCS involved in strategic management and product management (see Figure 4). 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. There are sub-groups even within the strategic formulators and strategy realizers groups. However, we considered the two groups as whole so that we could perform comparison between products since all groups do not have same sub-groups. In addition to the alignment of views between the two groups, internal agreement (between strategy realizers) is evaluated to ensure that information is not lost.
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. Before starting the evaluation, it is important to explain the purpose of the evaluation to gain trust of the SCS. Since it is very sensitive and confidential information, they need to be ensured of confidentiality and anonymity.

While performing evaluations, it might be revealed that the sample of chosen stakeholders is not enough and there is a need to perform these evaluation sessions with more SCSs. In that case re-sampling should be done to ensure as complete coverage of the SCSs as possible.

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 [37]. 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SCS1</td>
<td>To be the leader</td>
<td>Focus on secondary competitor</td>
<td>Product strategy document contains statements about vision and mission</td>
<td>We do not measure progress towards the goals</td>
<td>Innovation should be explicitly considered</td>
<td>Should not be considered in a product’s strategy rather it is the responsibility of development department</td>
<td>Is informally considered</td>
<td>There are product roadmaps that are followed</td>
</tr>
<tr>
<td>SCS2</td>
<td>To be the leader</td>
<td>Focus on secondary competitor</td>
<td>Product strategy document contains statements about vision and mission</td>
<td>We do not measure progress towards the goals</td>
<td>We do not focus on innovation thus it should not be explicitly considered</td>
<td>Should be stated explicitly in a product’s strategy</td>
<td>Not considered at all</td>
<td>Roadmaps are followed to a certain extent</td>
</tr>
<tr>
<td>Views</td>
<td>Similar</td>
<td>Similar</td>
<td>Similar</td>
<td>Similar</td>
<td>Not similar</td>
<td>Not similar</td>
<td>Not similar</td>
<td>Almost similar</td>
</tr>
</tbody>
</table>
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.

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>
<thead>
<tr>
<th>Results</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>It has been observed in the study results that on innovation perspective, the SCSs did not agree (taken from hypothetical example shown in Table )</td>
<td>Should innovation be explicitly considered or not in the product strategy?</td>
</tr>
<tr>
<td>Some think it should be considered explicitly and some think it should not be</td>
<td>Yes/No</td>
</tr>
<tr>
<td></td>
<td>Why do you think it should/should not be considered in the products strategy?</td>
</tr>
<tr>
<td></td>
<td>What in your view is the reason for this misalignment?</td>
</tr>
</tbody>
</table>
Table 7 – Follow-up questionnaire template for questionnaire results

<table>
<thead>
<tr>
<th>Results</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation matrix showing the degree to which the groups are aligned in how they perceive the usage of strategy goals for Product A</td>
<td></td>
</tr>
<tr>
<td>Correlation matrix showing the degree to which the groups are aligned in how they perceive the priority of strategy goals for Product A</td>
<td></td>
</tr>
<tr>
<td>Correlation matrix showing the degree to which the groups are aligned in how they perceive the priority of strategy goals for Product A</td>
<td></td>
</tr>
<tr>
<td>Overall prioritization of the five strategic factors according to their relative importance currently (from most important to least important).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stakeholders</th>
<th>SCS1</th>
<th>SCS2</th>
<th>SCS3</th>
<th>SCS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCS1</td>
<td>XX</td>
<td>YY</td>
<td>ZZ</td>
<td></td>
</tr>
<tr>
<td>SCS2</td>
<td>XY</td>
<td>YZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCS3</td>
<td>XZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCS4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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….
8. Any other….

Do you agree A is the most important factor?  
Yes/No

Why?

Do you agree D is the least important factor?  
Yes/No

Why?
Overall prioritization of the five strategic factors according to their relative importance that should be ideally (from most important to least important)

<table>
<thead>
<tr>
<th>1. Factor B</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Factor C</td>
</tr>
<tr>
<td>3. Factor D</td>
</tr>
<tr>
<td>4. Factor A</td>
</tr>
<tr>
<td>5. Factor E</td>
</tr>
</tbody>
</table>

Questions

<table>
<thead>
<tr>
<th>Do you agree B should be most important factor?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ] Yes</td>
</tr>
<tr>
<td>[ ] No</td>
</tr>
<tr>
<td>Why?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Do you agree E should be the least important factor?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ] Yes</td>
</tr>
<tr>
<td>[ ] No</td>
</tr>
<tr>
<td>Why?</td>
</tr>
</tbody>
</table>

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 [20-25, 38]:

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 [38]. 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 [38], 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 reflect 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). The purpose of this case study is to demonstrate practical application of MASS for alignment assessment and root cause identification.

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. The products’ characteristics are given in Table 8. It is
important to note that the findings through MASS application are specific to the products considered and thus cannot be generalized for all Ericsson’s products.

<table>
<thead>
<tr>
<th>Products</th>
<th>Size</th>
<th>Maturity (in terms of years)</th>
<th>Total releases</th>
<th>Release frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Large</td>
<td>9</td>
<td>7</td>
<td>1 per year</td>
</tr>
<tr>
<td>Y</td>
<td>Small</td>
<td>4</td>
<td>1</td>
<td>1 per 4 years</td>
</tr>
<tr>
<td>Z</td>
<td>Medium</td>
<td>9</td>
<td>4</td>
<td>1 per 2 years</td>
</tr>
</tbody>
</table>

### 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 3.1. 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. The objective was to catch these types of misalignments in perceiving the importance of each goal without any pre-set categorization of critical and not so critical goals. The documentation analysis and design of evaluation framework took 30 hours in total for the primary author i.e. roughly 4 person-days. Since the documentation of three products was read and analyzed, it can be said that it took 10 hours per product. Moreover, the evaluation framework was designed in a way to elicit maximum information from the participants without expending expensive industry resources.

While doing documentation analysis, one could be easily caught into “analysis paralysis”, therefore it is important to take help of the experts within the product to identify relevant documentation. Otherwise too much time can be spent on a lot of documentation which might be even relevant. Initially the authors were overwhelmed by the amount of documentation available however, then they sought help from the experts to identify the relevant documentation only.

#### 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. These roles were identified through brainstorming sessions with experts at Ericsson. For each product there was an 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).
The brainstorming session took an hour to identify roles and persons to be
interviewed within those roles.

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.

The interviews were designed to take one hour per interviewee thus in total it took
six hours for the primary author to conduct all the interviews.

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>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Similar view</td>
<td>Similar view</td>
<td>Similar view</td>
<td>Not similar view</td>
<td>Not similar view</td>
<td>Similar view</td>
<td>Similar view</td>
<td>Similar view</td>
</tr>
<tr>
<td>Y</td>
<td>Similar view</td>
<td>Similar view</td>
<td>Similar view</td>
<td>Almost Similar view</td>
<td>Not similar view</td>
<td>Similar view</td>
<td>Similar view</td>
<td>Similar view</td>
</tr>
</tbody>
</table>
Looking at column two, three and four in Table 9 agreement between the UPMs (UPM: upper product manager) and SPMs (SPM: strategic product manager) 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: “An 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 (UPM: upper product manager) and the SPMs (SPM: strategic product manager) 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

<table>
<thead>
<tr>
<th>Products</th>
<th>Stakeholders</th>
<th>UPM</th>
<th>SPM1</th>
<th>SPM2</th>
<th>SPM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product X</td>
<td>UPM</td>
<td>-90.0</td>
<td>-10.3</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td>35.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM2</td>
<td></td>
<td>56.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Y</td>
<td>UPM</td>
<td>77.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Z</td>
<td>UPM</td>
<td>15.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Products</th>
<th>Stakeholders</th>
<th>UPM</th>
<th>SPM1</th>
<th>SPM2</th>
<th>SPM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product X</td>
<td>UPM</td>
<td>63.2</td>
<td>13.5</td>
<td>70.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td></td>
<td>41.0</td>
<td>56.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM2</td>
<td></td>
<td></td>
<td>76.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Y</td>
<td>UPM</td>
<td>73.8</td>
<td>60.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Z</td>
<td>UPM</td>
<td>63.2</td>
<td>80.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 (UPM: upper product manager) and SPMs (SPM: strategic product manager), 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.

<table>
<thead>
<tr>
<th>Products</th>
<th>Stakeholders</th>
<th>UPM</th>
<th>SPM1</th>
<th>SPM2</th>
<th>SPM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product X</td>
<td>UPM</td>
<td>63.2</td>
<td>13.5</td>
<td>70.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td>41.0</td>
<td>56.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM2</td>
<td>76.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Y</td>
<td>UPM</td>
<td>73.8</td>
<td>60.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td>94.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Z</td>
<td>UPM</td>
<td>63.2</td>
<td>80.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SPM1</td>
<td>58.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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: “no 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 5 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.
The follow-up workshop took two hours. It is very important to present all the results clearly and precisely in order to conduct an effective and efficient follow-up session. It was personally experienced that since the results were clear and explicit the participants quickly got into the discussions about the root causes. The participants highly valued the follow-up to dig deep down into the root causes. They saw it as a better starting point for identifying issues compared to direct jump to the solutions based on the results only.

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. [39]. The validity threats considered are conclusion, construct, 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 Construct Validity
Evaluation apprehension can affect the way participants responded to the alignment evaluation. In order to mitigate this threat, authors made it explicit to the participants that there are no wrong and right answers but rather the purpose is to elicit their
understanding, opinions and views regarding product strategies formulation in general and results in particular.

4.2.3 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.

4.2.4 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 (UPM: upper product manager) and SPMs (SPM: strategic product manager) 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 in identification of misalignments as well as the root causes. Ericsson understood if there are misalignments among involved SCS with respect to the main tool (i.e. product strategies) for taking product development and management decisions, challenges like initial triage of requirements, balancing the long-term with short-term goals, balancing functional requirements with non-functional aspects cannot be addressed until the identified root causes are dealt with. 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.

As a result of this evaluation, Ericsson has already started research initiatives as how to solve the root cause C3: lack of explicit explanation of strategic factors and their importance in documented product’s strategy. Currently at the time of writing this paper, Ericsson along with the authors is exploring the possibilities for explicit and common vocabulary of strategic factors. They plan to work further on the lines as how to make strategic decision-making explicit once the strategic factors are made explicit.

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 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 VMOIST 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 [20-25] 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


ABSTRACT

In software intensive products such as cars or telecom systems, software has traditionally been associated with cost, and there has been no real perception of its value in relation to the entire product offering. However, as software is becoming a larger part of the main competitive advantage, driving innovation and product differentiation, hardware is becoming more standardized, thus the valuation of software is becoming critical. In existing literature, several value components and corresponding valuation/measurement solutions needed for making decisions about software product development are presented. However, the contributions are often isolated with respect to a certain perspective such as focusing on product’s internal or external quality aspects only. Consequently, a complete view of value components relevant from different perspectives required for making decisions about software product development is missing. This paper presents a consolidated view of the software value concept utilizing the major perspectives and introduces a Software Value Map. The created value map was evaluated through an industry case study through the development of Impact evaluation patterns which were subsequently used by industry professionals in industry, and experiences gathered. During industry evaluation, practitioners found substantial benefits of having a consolidated, vastly improved and extended value aspect’s view of software.

Keywords
Value-based software engineering, decision-making, software value, software value analysis, Software Value Map, requirements engineering, technology and software product management, product customization, software engineering management.

1. INTRODUCTION

It is very hard to estimate and calculate software value, and this is especially true for embedded or hardware intensive products. In many cases, the traditional view has been that the software part of a product was the “poor cousin” that “had to be there”, bundled with the hardware, but without any real value in itself [1, 2]. Examples of this can be seen in many embedded fields like the automotive domain or in the automation domain. Thus, at best software was handled as a cost and there was no real perception of the software value beyond immediate sales of the product [3, 4].

In today’s competitive world, however, software has become the main competitive advantage, enabling faster and cheaper innovation as well as product differentiation, and at the same time hardware is becoming standardized [5]. Simultaneously the size and complexity of software in products are increasing, and so is the impact of software development decisions on the overall product offering. That is, any decision taken regarding software (e.g. what features to realize, what quality to offer, or what technology to choose) will impact the entire product’s life cycle and value, not to mention that it limits future possibilities and direction of the product and business [2, 6, 7]. This situation gives rise to many decision-making challenges for industry practitioners, for example, what is the actual value of software? How does the realization of one feature or quality aspect influence the overall value of the product offering, where short-term potential sales and revenues are only small parts of the complete picture?

Practitioners discussing value often talk about either different things, or have too shallow understanding/view on the subject. This can be attributed to the lack of common definitions and interpretations of the concept of software value. This also leads to the fact that short-term gains, for example revenues or immediate customer satisfaction, are the dominant value aspects taken into consideration, while other aspects such as architectural impact or long-term customer lock-in, commissioning efficiency, and so on go largely ignored. A common understanding of the concept of value, as well as a way in which this complex value aspects can be used in daily decision making are central in the development of software intensive systems.

Value-based software engineering (VBSE) emphasizes that every decision and/or feature of a product does not have an equal value like in a value-neutral setting [2]. This requires making decisions that are better for overall value creation [8, 9], and balancing short-term and long-term value creation. In order to enable this, different perspectives and
corresponding value aspects need to be considered together while making decisions for sustaining growth and maintaining the competitive advantage.

In relation to this the research presented in this paper offers four main contributions. First, a consolidated view (called the Software Value Map) of the software value concept is presented utilizing four major perspectives, (i) financial, (ii) customer, (iii) internal business process, and (iv) innovation and learning. The value map offers a unified view of value where value concepts are categorized as value aspects, sub-aspects and value components, which can be used by professionals to develop a common understanding of value, as well as acting as decision support to assure no value perspective is unintentionally overlooked when taking product management and development decisions. The second contribution is the mapping of links between different value aspects - making interrelationships explicit. As a part of this categorization, we also collected measures, measurement methods, techniques and models (called measurement solution from here on in) used to measure a specific value component. As a third contribution, the gaps are mapped. That is, missing value aspects and value components, and connections between different value aspects and components are made explicit to lift the importance of future research to fill the gaps. Fourth but not least, the created Software Value Map (SVM) was used to construct a number of Impact evaluation patterns at Ericsson, in essence a tailoring of the value map to an industry case and evaluating the value map.

The research presented here is a result of collaborative research conducted with Ericsson AB Karlskrona. Ericsson is involved in the development of embedded software and, realizing the importance of value-based software product development and is currently evolving its advanced conceptualization of software as a central value component in its embedded products.

The paper is structured as follows. Section 2 gives an overview of value from different perspectives relevant for software product development organization based on software engineering, business, management and economics literature. Section 3 details the research methodology used to identify the need for the Software Value Map, its creation and evaluation in industry along with associated validity threats. The results of the literature review are presented in Section 4. Taxonomy, structure and definitions of the value map are presented in Section 5. In Section 6, the industry evaluation of the Software Value Map is presented together with Impact evaluation patterns created for the selected strategic decision-making activity at Ericsson. Lessons learned during this case study are presented in Section 6.4, and Section 7 concludes the paper.

2. BACKGROUND AND RELATED WORK

Research focusing on value-based software engineering (VBSE) recommends that the economic value perspectives be integrated into the software engineering processes to extend the traditional scope of software engineering from the technical issues to business relevant decision challenges [2]. Section 2.1 expands on the absence of a consolidated view of value and elaborates on associated repercussions. Section 2.2 provides a summary of the measurement and valuation solutions presented in literature for measuring different value components.

2.1 The Absence of a Consolidated View on Value

Several researchers have presented value components and corresponding valuation/measurement solutions needed for making decisions about software product development [9-14]. However, the contributions are often isolated and have a limited
perspective, focusing on e.g. only cost, or only product characteristics like usability. Consequently, a complete picture of value components, relevant from different perspectives required for taking software product management and development decisions, is missing. For example, from the product perspective, if software value is measured by measuring external and internal quality attributes [15], say usability or reliability, it is only one isolated measure. The usability value is often shown as a number (for example, understandability = A / B, where A = Number of functions (or types of functions) understood, and B = Total number of functions (or types of functions)). The question remains however, how does this number relate to the software product value relevant from different perspectives, like the overall customer perceived value, or the impact on internal business processes?

Similarly cost-benefit analysis methods like CBAM and LiVASAE [16, 17], based on subjective evaluation of architectural strategies with respect to quality attributes of the architecture, fall short on connecting cost-benefit discussions to broader context (e.g. customer perceived value, innovation value and or the impact on internal business processes).

Business research investigates value components from a number of other perspectives, i.e., production value, differentiation value, intellectual capital value and shareholders’ value [10, 18]. However; in software engineering literature, these have little or no explicit connection to the software product planning (requirements triage, requirements selection and product management), and/or development (design, coding and testing) activities [2, 19].

Further, the value of software is influenced by various ‘non-technical’ factors (for example, business and marketing) [20], and realized only after the product is marketed [14]. Moreover, many businesses, marketing and technical factors are interrelated [21]. For example, the perceived value of software is dependent on many factors like the product characteristics and marketing efforts. The product characteristics in turn, are determined by the development process characteristics like elicitation quality, verification and validation standards, but also by the system properties, like usability, performance, and reliability [19, 21]. When considering business, marketing and technical factors independently, there is also a risk of overlap or redundancy, which can affect the accuracy of valuation.

2.2 Measurement and Valuation Methods

While reviewing economics, business, management and VBSE literature, we collected measurement solutions used to measure/value a specific value component. This subsection gives a brief summary of measurement and valuation concepts. This will help readers understand the measurement solutions listed for the value components in the Software Value Map.

Measurement can be defined as “the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules” [22, 23]. A central goal of measurement is to capture the characteristics of entities and manipulate them in some formal way. Examples of software entities are products, processes, or resources of different types. Attributes are defined as the properties that an entity possesses. A measure maps an empirical attribute to the formal, mathematical world. A measurement unit determines how an attribute is measured. When measurement units are considered, the measurement scale type implied by the unit needs to be understood. The most common scale types are: nominal, ordinal, interval and ratio. It is fundamental to be aware of the unit’s scale types because it determines the admissible transformations that can be applied to particular units [24].

There are two broad types of measurement: direct and indirect. Direct measurement of an attribute is a measurement that does not rely on measurement of any other attribute. Whereas, indirect measurement of an attribute is measurement which entails the measurement of one
The valuation of software product/projects/processes depend on a detailed analysis of underlying costs and benefits. However, the valuation of software products and related activities is so far nearly an unexplored area in the software engineering context [3]. In economics, various valuation methods ranging from intuitive judgment to complex options models [14] have been developed and utilized. Although individual methods may differ from one another in terms of the criterion and procedures used, the “value” of software is articulated from different perspectives using score, index, or monetary value. Among others, scoring methods have been widely used. For example, while performing customer value analysis (CVA), a number of value components (performance, usability, usefulness) can be given to the evaluators (in this case the customers’ representative) and then evaluators can subjectively rate score for each factor for the product and other competing products. Similarly, this can also be done for innovative technology valuation. Scoring models are popular because they are simple and robust [14]. However, the score itself never conveys the real meaning of value; rather it indicates the relative preference among alternatives.

Alternatively, index models can be used to develop a functional form of valuation. This type of model is more flexible, as it can accommodate more diverse measures such as ratio and percent. But it is still a ratio or percentage, and fails to convey the actual value. On the contrary, monetary value model attempt to measure the monetary worth by using capital budgeting methods to estimate the discounted cash flow (DCF) to calculate net present value (NPV). For example, the traditional shareholder value model uses NPV calculations to calculate shareholder value [26].

Monetary value models can be further categorized into three basic approaches: cost approach, market approach, and income approach [26]. The cost approach is based on the economic principle of substitution that states a practical buyer would pay no more, and a willing seller can ask no more, for a product than the cost to create the intellectual asset of equal quality and utility. This approach has been applied mostly to calculate the cost of product development however; it only calculates one factor for determining value.

The market approach is a simple and direct method that argues the value of a product is equivalent to what others in the market place have judged it to be. Customers are potential judges and methods like CVA helps to calculate the monetary value of the software product however, as discussed earlier this approach is quite subjective. Additionally, in relation to embedded products the software is difficult to separate from the overall offering. Thus a central problem for valuation when using external sources for the value estimation is that they see the entire product and not just the software. This is not completely handled by the proposed Software Value Map, however; by combining internal value estimations the software part of the value can be gauged [2].

The income approach is based on the rationale that value is determined by income-producing capability of the product. The income approach is considered to be best suited for the valuation of intellectual property such as software, patents, trademarks and copy rights [14]. It might be desirable but not always easy to calculate monetary worth of every value aspect, for example, while determining user experience value, defined as “a consequence of a user’s internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g. organizational/social setting, meaningfulness of the activity, voluntariness of use, etc.)” [27], we have to resort to subjective scoring. Similarly, for estimating the value of usability or reliability we can use the measurements and metrics proposed in ISO 9126 [15].

The central challenge lays in the fact that different valuation models are used for valuing/measuring different value components, and from different perspectives, which results in incompatibility between the measured values. Based on an extensive literature study and
an industry case study (see Section 3), a list of challenges was compiled. These challenges summarized in Table 1 stand as motivation for the creation of the SVM.

### Table 1 - Challenges in decision-making

<table>
<thead>
<tr>
<th>C1: Multiple perspectives/views and lack of common vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>It was found during exploratory studies at Ericsson that practitioners in the product management and development have different definitions and understanding of value aspects and they do not consider all the value aspects while making decisions. Decision-making, on the other hand, often involves deliberations in different perspectives. Distinct perspectives or views support knowledge acquisition and representation suitable for different types or stages of inference in the same discourse. Involvement of distinct perspectives does offer spectacles for looking at a decision from different aspects however; this gives rise to misinterpretations and misunderstandings about value aspects and value components to be considered.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C2: Long-term versus short-term gains and losses</th>
</tr>
</thead>
</table>
| 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 [28]. 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 [29]. 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):
  1. First, large quantities of requirements, sometimes numbering in the thousands or even tens of thousands, risk to overload the development organization [30], thus initial triage of requirements is necessary [31, 32].
  2. 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.
  3. 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 [33]. |

In this environment, long-term costs and benefits in terms of sustainability and innovation are often ignored or overlooked. Practitioners decide what to include in a product by looking only at the short-term costs, short-term customer satisfaction and short-term sales. However, sustainable software (sustainability: a term used interchangeably with software maintenance [34]), and an innovation infrastructure are fundamental inputs to continuously maintain and evolve software products such that they remain competitive throughout their planned lifecycle.

<table>
<thead>
<tr>
<th>C3: Release planning quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commonly, software intensive product developing organizations provide successive releases of the product and release planning is a critical activity [35]. A major challenge in MDRE is to prioritize and select the right set of requirements to be implemented in the next release [36], while avoiding obstruction in the selection process [37]. This decision-making is very challenging as it is based on uncertain predictions of the future, while crucial for the product’s success on the market [35].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C4: Value-based impact and consequence analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Money is the least common denominator that can be compared across all enterprise functions; however, financial/economic view of a software product life cycle is missing today. This limits the possibility to compare actual value of adding and maintaining a feature in software for detailed analysis with respect to total cost of ownership, packaging and procurement. Additionally, impact and consequences on sustainability and innovation potential are ignored.</td>
</tr>
</tbody>
</table>

Each of these challenges is addressed directly through the creation and use of the Software Value Map as can be seen in the remainder of the paper, and explicitly discussed in Section 6.4.
3. RESEARCH METHODOLOGY

The general research approach used for the creation of the Software Value Map and its validation in industry is shown in Figure 1. The need for the creation of Software Value Map was identified through exploratory case studies at Ericsson (Step 1).

![Figure 1 - Overview of research approach](image)

Based on the problem statement, an extensive literature review was undertaken to identify state-of-the-art (Step 2, see Section 3.1 for details). Through this literature review, the Software Value Map was created as a candidate solution for the problems identified (Step 3, see Section 5 for details). Since the Software Value Map was created based on needs identified in industry, a natural progression was an evaluation in industry. Through collaboration with Ericsson AB, static evaluation of the value map concept was performed through creation of Impact evaluation patterns and use of the created patterns for decision-support (Step 4, see Section 6).

3.1 Step 2: Literature Review Methodology

Extensive literature studies were used to identify all the value aspects and value components relevant for making decisions in the development of software intensive products. Ideally a systematic mapping of all the relevant literature in economics, management, business and software engineering should have been performed. However, since it was not practical to cover all literature from so many subject areas, as a first step a systematic mapping was performed to collect all value-based aspects from software engineering literature. Then as a next step, snow-ball sampling was used to cover all the areas other than software engineering to identify value aspects and value components that could be relevant. The design of each study is described below.

3.1.1 Step 2.1: Systematic Mapping

The systematic mapping methodology outlined in [38, 39] was used. The aim was to find all the value aspects, sub-aspects and value components that are or should be considered while making decisions related to the development and management of software intensive
Search Strategy and Screening

Search strings were formulated through the following steps.
- All possible value keywords were identified by scanning research papers’ title, abstract, and index terms (research papers referenced in Section 2 were used as a starting point)
- Main terms were identified by determining the population and the intervention
- Synonyms of the keywords were identified by using a thesaurus
- Last, search strings were formulated using boolean operators such as AND, OR, PRE, ONEAR etc

In the search strings all the phases of software development were included to identify all value aspects relevant and/or used in different phases of software development. Exact keywords used and the search strings are available online [40]. Table 2 gives an overview of the data items relevant for the systematic mapping.

Table 2 - Systematic map search strategy

<table>
<thead>
<tr>
<th>Data Items</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases searched</td>
<td>Inspec and Compendex (through Engineering Village) and Scopus</td>
</tr>
<tr>
<td>Population</td>
<td>Software development process areas AND Decision-making</td>
</tr>
<tr>
<td>Intervention</td>
<td>Value related keywords</td>
</tr>
<tr>
<td>Outcomes</td>
<td>The outcome of systematic mapping study was the systematic classification and mapping of the research papers with reference to value aspects.</td>
</tr>
<tr>
<td>Search queries formulation</td>
<td>To make the search exhaustive, electronic databases were searched using the following strategy</td>
</tr>
<tr>
<td></td>
<td>- Boolean OR was used in between the interventions.</td>
</tr>
<tr>
<td></td>
<td>- Boolean AND was used in between the population and intervention.</td>
</tr>
<tr>
<td></td>
<td>- In order to restrict the search to research papers that contain only value aspects and the term &quot;software&quot; boolean AND was also used in between them.</td>
</tr>
<tr>
<td></td>
<td>Detailed queries are available on the weblink [40]</td>
</tr>
<tr>
<td>Out of scope</td>
<td>Fields such as enterprise resource planning, computer aided software engineering, web services etc, were out of scope of this study. The rationale being the focus was on the value considerations reported in software engineering literature.</td>
</tr>
<tr>
<td>Reference management system</td>
<td>Endnote.</td>
</tr>
<tr>
<td>Year</td>
<td>The research papers are selected from the year 1969 to 2010.</td>
</tr>
<tr>
<td>Papers Targeted</td>
<td>Research papers published in peer-reviewed journals, conference and workshops. Editorials, prefaces, summaries, news, reviews, correspondence, discussions, comments, reader’s letters and summaries of tutorials, workshops, panels, and poster sessions were also excluded.</td>
</tr>
</tbody>
</table>

Using the search strings [40] in Stage 1 (see Figure 2), a total of 670 papers were retrieved. In Stage 2 the duplicates were removed leaving 558 unduplicated papers. For each of the 558 papers, the source, the retrieval decision, retrieval status, and eligibility decision were recorded.
In Stage 3, the titles of all papers were screened to judge their relevance to the systematic mapping being performed. The papers whose titles were clearly not related to software engineering and value were excluded. In Stage 3, 417 papers were excluded leaving 141 papers in total. In Stage 4, 87 papers out of 141 were excluded primarily after reading the abstracts leaving 54 papers. The reason for excluding the 87 papers was absence of any value considerations for decision-making. 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 a paper included any value aspects. If it was unclear from the title, abstract, conclusion and keywords whether a research paper conformed to the screening criteria, it was included in the mapping, giving it the benefit of the doubt.

The inclusion and exclusion criteria were piloted by the authors on a random sample of 20 papers. The Fleiss’ Kappa [41] value showed a reasonable agreement (0.67) among the authors.

Establishing a Classification Scheme

Since the purpose of the systematic mapping was to identify which value aspects and value components were considered while taking decisions, the value keywords [40] from the papers were used to classify the papers. Papers stating similar value aspects with different names, for example “product value” and “exchange value” were classified under one value aspect (in this case “product value” was used in the classification). The definitions/descriptions stated by the authors in their papers were used to judge similarity and was used for the classifications.

3.1.2 Step 2.2: Snowball Sampling

Snowball sampling is a non-probability sampling technique [42]. It involves starting with the review of one or two seminal research papers and/or books and pursuing references of references and electronic citation tracking [43]. This gives rise to an external validity threat with respect to the reliability of literature review [44]. The reliability has been addressed as
far as possible by starting the snowball sampling by going through the reference lists of all
the papers selected for systematic mapping for example. As a result, 13 additional papers
were identified. The value aspects identified in both the systematic mapping and the
snowball sampling are detailed below. Complete set of papers and value aspects are available
online [40].

### 3.2 Threats to validity

#### 3.2.1 Publication bias

Publication bias refers to the general problem that positive research outcomes are more
likely to be published than negative ones [45]. This is considered as a minor threat, since the
research questions in this mapping are not geared towards the performance of a specific
solution for the purpose of a comparison. The same reasoning applies to the threat of
sponsoring in which certain methods are promoted by influential organizations [46], and
negative research outcomes regarding this method are not published. The sources of
information were not restricted to a certain publisher, journal or conference such that it can
be assumed that the breadth of the field is covered sufficiently. However, there is a trade-off
between considering as much literature as possible and, at the same time, accumulating
reliable information. Therefore grey literature (technical reports, work in progress,
unpublished or not peer-reviewed publications) was excluded [46].

#### 3.2.2 Threats to the identification of primary studies

Only two databases were searched (Scopus and EngineeringVillage), due to the fact that
Scopus has relatively lesser weaknesses compared to databases such as IEEEXPlore, ACM
Digital Library, SpringerLink, ISI Web of Knowledge and ScienceDirect [47]. Additionally
EngineeringVillage was also searched to retrieve as many papers as possible.

Another potential threat related to the identification of primary studies is that among the
selected studies some have been written by the authors’ colleagues. However, the selected
studies were the ones identified through the developed search strategy (which was reviewed
by an external person) thus this threat is not huge for the selection.

#### 3.2.3 Threats to selection and data extraction consistency

Due to the scope of the systematic mapping, there was a need to develop efficient (in
terms of execution time) and effective (in terms of selection and data extraction consistency)
strategies. One of the main aims of defining a mapping protocol is to reduce researcher bias
[46] by defining explicit inclusion/exclusion criteria and a data extraction strategy. A well
defined protocol increases the consistency in selection of primary studies and in the
following data extraction if the mapping is done by multiple researchers. One approach to
further increase the validity of the mapping results is to conduct selection and data extraction
in parallel by several researchers and then crosscheck the outcome after each phase. In the
case of disagreements they should be discussed until a final decision is achieved. Due to the
large amount of initially identified studies (558) it was impossible to implement this strategy.
Therefore, as proposed by Brereton et al. [48] paper selection and data extraction were
piloted and the consensus was improved iteratively. Two issues were addressed through
piloting: first, the selection criteria and the data extraction form were tested for
appropriateness, and second, the agreement between the researchers could be assessed and
discrepancies streamlined. Although it can be argued that this strategy is weaker in terms of
consistency than the previously mentioned crosschecking approach, it was a necessary trade-
off in order to fulfill the schedule and the targeted breadth of the systematic mapping.
4. LITERATURE REVIEW RESULTS

This section presents results of literature review (Step 2, see Figure 1). Table 3 gives an overview of the value aspects identified as well as a short description of each aspect and the corresponding sources (the sources, e.g. P1, P2..Pn) can be found online [40].

Several contributions have been made integrating economic and value perspectives. Aurum and Wohlin [49] emphasized that adding value is an economic activity that has to be taken into account from a business perspective. Value is created when a company makes a profit by succeeding in the market. Thus, the critical success factor for software companies is their capability to develop and deliver a product that satisfies customer requirements while offering high value that provides increased support for market success [2]. Thus, value is created for the customer [P3, P10, P30, P33, P35, P36, P42, P43, P49, P50 and P60]. In this context, it appears logical to consider customer value as a strategic merit to assess the value of a product and also to assess the overall value of business [21].

Table 3 - Summary of results of systematic mapping and snowball sampling

<table>
<thead>
<tr>
<th>Value aspect</th>
<th>Description</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers perspective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perceived Value (PV) - use value</td>
<td>Benefits derived from the product/feature. It is the trade-off between perceived benefits and the cost of ownership.</td>
<td>P21, P27, P39</td>
</tr>
<tr>
<td>Intrinsic value</td>
<td>It is embedded into the software as functionality and attributes e.g. usability, security.</td>
<td>P28, P31, P34, P55</td>
</tr>
<tr>
<td>Functionality</td>
<td>The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.</td>
<td>P55</td>
</tr>
<tr>
<td>Reliability</td>
<td>The capability of the software product to maintain a specified level of performance when used under specified conditions.</td>
<td>P55</td>
</tr>
<tr>
<td>Usability</td>
<td>The capability of the software product to be understood learned, used and attractive to the user, when used under specified conditions.</td>
<td>P55</td>
</tr>
<tr>
<td>Maintainability</td>
<td>The capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.</td>
<td>P55</td>
</tr>
<tr>
<td>Portability</td>
<td>Measures such attributes as the behavior of the operator or system during the porting activity</td>
<td>P55</td>
</tr>
<tr>
<td>Delivery process value</td>
<td>Quality of process in installing/upgrading/receiving the product</td>
<td>P56</td>
</tr>
<tr>
<td>Network externalities</td>
<td>The amount of other users of the software product that are relevant to the focal user</td>
<td>P18, P44, P53, P68</td>
</tr>
<tr>
<td>Complementary value</td>
<td>A complement can be defined as a product/service, which increases the value of another product/service so complementary value refers to value, which is created by combining a piece of software with another good/service</td>
<td>P44</td>
</tr>
<tr>
<td>User experience value</td>
<td>A factor in addition to Usability, which determines success or failure on the competitive market</td>
<td>P57-P59</td>
</tr>
<tr>
<td>Customer lifetime value</td>
<td>Represents a profound supplier-oriented understanding of customer value which is measured as profit streams of a customer across the entire customer life cycle</td>
<td>P60</td>
</tr>
<tr>
<td>Value aspect</td>
<td>Description</td>
<td>References</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Retention rate</td>
<td>Is a factor which is typically defined with regard to the individual customer that he/she remains loyal to a particular supplier and keeps yielding expected revenues as well as costs within a fixed period of time</td>
<td>P60</td>
</tr>
<tr>
<td>Revenue</td>
<td>Profit, money made</td>
<td>P60</td>
</tr>
<tr>
<td>Costs</td>
<td>Costs incurred</td>
<td>P60</td>
</tr>
<tr>
<td><strong>Internal business perspective</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production value</td>
<td>Represents an aggregated value of software production process, which may involve its physical value (PV) and its market purpose and values</td>
<td>P63</td>
</tr>
<tr>
<td>Market requirements value</td>
<td>Represents the production value with respect to a given market requirement</td>
<td>P63</td>
</tr>
<tr>
<td>Physical (project) value</td>
<td>Represents the integration of both product's features and software process deployment for better organizing critical complexities within the production system. Used to estimate the real added value of production processes and final product and its influence on markets</td>
<td>P5, P23, P25, P41, P47, P48, P63</td>
</tr>
<tr>
<td>Physical value related to Time (PVt)</td>
<td>Represents the PV wrt Time. A product being evaluated, adjusted and put into market within a shorter time production cycle will have higher PVt</td>
<td>P63</td>
</tr>
<tr>
<td>Physical value related to Quality (PVq)</td>
<td>A company that has more flexible organizational structure to deal with virtual development processes will have higher PVq - organization</td>
<td>P63</td>
</tr>
<tr>
<td>Physical value related to Quality of Process (PVq-process)</td>
<td>A company that uses &quot;industry best practices&quot; for process improvement and quality assessment will have a higher PVq-process</td>
<td>P63</td>
</tr>
<tr>
<td>Physical value related to Quality of product (PVq-product)</td>
<td>A product that exposes functionalities with higher quality benefit according to industry standards will have higher PVQ-product</td>
<td>P63</td>
</tr>
<tr>
<td>Architectural value</td>
<td>Related to the architecture and design aspects</td>
<td>P11, P51, P55</td>
</tr>
<tr>
<td>Physical value related to Cost (PVc)</td>
<td>A product being developed and marketed with lower production cost will have higher PVc</td>
<td>P63</td>
</tr>
<tr>
<td>Differentiation value</td>
<td>The differences of a product or offering from others, to make it more attractive to a particular target market. This involves differentiating it from competitors' products as well as one's own product offerings</td>
<td>P64</td>
</tr>
<tr>
<td><strong>Financial perspective</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shareholder’s value</td>
<td>Is determined by estimating the economic value of an investment by discounting forecasted flows by the cost of the capital</td>
<td>P61, P6</td>
</tr>
<tr>
<td><strong>Innovation and learning perspective</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value of technology</td>
<td>The potential value that is embedded in the subject technology itself</td>
<td>P8, P67</td>
</tr>
<tr>
<td>Value of market</td>
<td>Practical value of subject technology that is materialized in market or in business process</td>
<td>P67</td>
</tr>
<tr>
<td>Intellectual capital value</td>
<td>Software product and process knowledge</td>
<td>P50, P65, P66</td>
</tr>
</tbody>
</table>
From the customer’s perspective, Rönkkö et al. [P44] proposed a value decomposition matrix based on the concept of value and utility in software business research and value of software as technology. They presented three value components, namely, intrinsic value, externalities [P68], and complementary value [P68]. Intrinsic value has been defined as the value that is embedded in the software as functionality and attributes such as security and reliability [P44]. Complementary value has been defined as a value, which is created by combining a piece of software with another good or service [P44]. If the software is used for communication, it is subject to network externalities and its value is dependent on the amount of other users of the software, relevant to the focal user. Perrey et al. suggested delivery process value as one of the value components to be considered when measuring customer perceived value [P56]. Several others discussed that user experience is another fundamental value component, among others, which contributes to the success or failure of a software product in a competitive market [P57-P59]. Exploring the customer perceived value definition further; one can see that the customer lifetime value concept represents a profound supplier-oriented understanding of customer value [P60]. Customer lifetime value is measured as profit streams of a customer across the entire customer life cycle. It is an application of contemporary finance to the evaluation of customer retention [P60]. However, in the existing software engineering literature this has not been considered when determining the value of a software product.

While considering the internal process perspective, Lei et al. [P63] presented a value model tree to address the common challenges faced by small-to-medium sized companies when entering the international market. The proposed value model tree specifies general relationship decomposition from final value, to market inputs and critical elements of the production processes, as well as human values related to risk attitude and time preferences. The production value represents an aggregated value of the software production process, which may involve its physical value and its market purpose and values [P63]. The proposed value model helps to visualize the internal production value with respect to the market requirements, production processes (project), and marketing activities. In addition, several other authors have discussed project value while taking decisions e.g. [P5, P23, P25, P41, P47, P48, P63].

From the financial perspective, shareholder value is a fundamental aspect for business valuation, in addition to customer perceived value and internal processes value, and needs to be considered while decomposing software value from the company perspective. In financial and business literature, increasing shareholder value was the central theme for the last two decades [P61, P62]. As a result, many organizations aimed at maximizing shareholder value and were constantly looking for new opportunities to maximize earned value added, or EVA, which led to the pursuit of attaining short-term objectives and targets [P61]. For example, prime focus had been on continuous increase of quarterly profits by delaying needed investments. Considerably less attention was given to long-term objectives, like investing in effective and efficient development processes, innovation, the quality of employees and the workplace. However, this approach makes sustainable growth hard to maintain and leaves almost no room to increase customer satisfaction with new innovative products or services [P61]. As a result, this approach affects the customer perceived value of the software product and needs to be considered when drawing a complete picture of software value. However, this aspect has not been discussed explicitly in VBSE literature.

In the current emerging competitive environment, product and service innovations differentiate a company from its competitors; technology valuation is gaining popularity [P67]. Thus, there has been a realization that the value of a corporation and business cannot be gauged without knowing the value of the technological assets [P67]. In the absence of knowledge about maturity, life cycle, contribution ratio, scope of application and standardization of the technology, the measurement/valuation of innovation with respect to
technology would be deceiving and irrelevant. Consequently, any product development based on a specific technology would affect the perceived value of the product. Others have discussed the value of intellectual capital (IC) as a fundamental value aspect to be considered for achieving long term prosperity and sustainability see e.g. [P50, P65, P66].

All of the value aspects identified during the literature reviews were used as input to creating the Software Value Map presented in Section 5. Although the focus of systematic mapping was to extract value aspects and value components from the selected papers, however Table 4 shows the top five venues where most of primary studies have been published.

<table>
<thead>
<tr>
<th>Publication venue</th>
<th>Number of papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Software</td>
<td>5</td>
</tr>
<tr>
<td>International Workshop on the Economics of Software and Computation</td>
<td>3</td>
</tr>
<tr>
<td>EUROMICRO Conference on Software Engineering and Advanced Applications</td>
<td>3</td>
</tr>
<tr>
<td>International Workshop on Software Process Simulation and Modeling</td>
<td>2</td>
</tr>
<tr>
<td>Communications of the ACM</td>
<td>2</td>
</tr>
</tbody>
</table>

5. Step 3: SOFTWARE VALUE MAP

The results from the literature studies were used to create the Software Value Map (see Figure 1, Step 3). The Software Value Map (SVM) shows a consolidated view of the value concept, presenting “true value” as a complex interaction of perspectives and views. This section presents the SVM, and the underlying concepts, structure and taxonomy, to categorize and describe value components.

5.1 Taxonomy

The taxonomy used to categorize the value aspects and value components was inspired by balanced scorecard (BSC) [50, 51]. BSC can be defined as a set of measures that gives managers a fast but comprehensive view of the business using four main perspectives, namely, the financial, customer, internal process, and innovation and learning, each described below [50, 51].

The financial perspective contains aspects that address the company’s implementation and execution of its strategy which are contributing to the bottom-line improvement of the company. It represents the long-term strategic objectives of the organization and thus it incorporates the tangible outcomes of the strategy in traditional financial terms [50, 52]. Some of the most common financial measures that are incorporated in the financial perspective are earned value analysis, revenue growth, costs, profit margins, cash flow, net operating income, and customer value analysis.

The customer perspective defines the value proposition that the company will apply to satisfy customers and thus generate more sales to the most desired (i.e. the most profitable) customer groups [52]. Measures that are selected for the customer perspective should measure both the value that is delivered to the customer (value proposition) with respect to the perceived value, which may involve time, quality, performance and service, and cost, and the outcomes that come as a result of this value proposition (e.g., customer satisfaction and market share).

The internal process perspective is concerned with the processes that create and deliver the customer value proposition. It focuses on all the activities and key processes required in order for the company to excel at providing the value expected by the customers both
productively and efficiently. These can include both short-term and long-term objectives as well as incorporating innovative process development in order to stimulate improvement. Quality, cycle time, productivity and cost are some aspects where performance value can be measured [50].

The innovation and learning perspective is the basis of any strategy and focuses on the intangible assets of an organization, mainly on the internal skills and capabilities that are required to support the value-creating internal processes. The innovation and learning perspective is concerned with the intellectual capital categorized as human capital, structural capital, and the organization capital of a company [50, 53].

BSC combines the financial perspectives with operational perspectives to provide a consolidated view of the business performance and was thus considered as a good base for the categorization in the creation of the Software Value Map.

5.2 Structure and Definitions

Several different value aspects and value components were identified, each needs to be specified and placed. Thus, the central challenge while creating SVM was to categorize and present all the value aspects and value components in a clear and structured manner. A nested structure with each value perspective (VP) being the parent node was chosen (based on concept-centric approach [54]). Each VP has nested clusters of nodes called value aspects (VA), and in turn a VA can have sub-value aspects (SVA). The end nodes are called value components (VC). Each concept is detailed below along with attributes specified for each. Table 5 shows the corresponding attributes specified for every VP/VA/SVA/VC.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Value perspective (VP)</th>
<th>Value aspect (VA)</th>
<th>Sub-value aspect (SVA)</th>
<th>Value component (VC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Name</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Description</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Motivation/rationale</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>References</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Additional notes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Composed of</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Measured through</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

The description of attributes is given below:

1) **ID:** Every VP has an ID prefixed with VP X where X = 1, 2, 3, …..n. Every VA has an ID prefixed with VA X.Y where X = 1, 2, 3, …..n of the parent VP, Y = 1, 2, 3, …..m. Every SVA has an ID prefixed with SVAX.Y.Z where X and Y are the same as above Z = 1, 2, 3, …..p. Every VC has an ID prefixed with VC X.Y.Z where X = 1, 2, 3, …..n of the parent VP, Y = 1, 2, 3, …..m of the parent VA (or SVA) and Z = 1, 2, 3, …..p

2) **Name:** The name should reflect the perspective/aspect/sub-aspect from which the value is categorized. For example, *Value for Customer* is one VP that categorizes value aspects and value components relevant from the customers’ perspective. The name should not be longer than five words.

3) **Description:** The VP, VA, SVA or VC description should not be longer than five
sentences, and should describe the central essence of the aspect in broad strokes.

4) **Motivation/rationale:** This attribute should specify the motivation in the context of the perspective being stated. If the perspective is of *customer* it should illustrate the motivation/rationale from the customers’ perspective.

5) **References:** This attribute specifies the references in literature which have introduced/addressed the VP/VA/SVA/VC.

6) **Additional notes:** This attribute contains any additional information about the VP/VA/SVA/VC.

7) **Composed of:** A mapping of what VAs are included in the VP, what SVAs are included in the VA and so on.

8) **Measured through:** contains a list of measurement solutions, presented in literature or otherwise, as to how to measure a VC. It contains a short description of the measurement solution and references to the literature where the measurement solution has been presented. However, this attribute does not contain an exhaustive list of measures that have been proposed in literature to measure/evaluate the corresponding VC.

**Value Perspectives (VP), Value Aspects (VA) and Sub-value Aspects (SVA):** Figure 3 illustrates the structure where each value perspective (VP) holds a cluster of related value aspects (VA). The VA can recursively have a cluster of SVAs. **Value Component:** A value component (VC) is the end node, i.e. the leaves in the model tree structure (see Figure 3 and Figure 4). It represents a discrete value factor that is measurable. Every child node (whether it be a VA, SVA or VC) inherits context from the parent node. The complete SVM, and user manual with a step-by-step instructions for pattern creation can be found online [40].
Looking at Figure 3; “Customer perspective” is one of four VPs (with seven attributes of its own). It contains two VAs: “VA1.1: Perceived value” and “VA1.2: Customer lifetime value”. Every VA also has seven attributes. Each VA further contains one or more SVAs or VCs. For example, VA1.1 contains three SVAs and two VCs namely: “SVA1.1.1: Intrinsic value”, “SVA1.1.2: Delivery process value”, “VC1.1.3: Network externalities”, “VC1.1.4: Complimentary value” and “SVA1.1.5: User experience value”. Looking further down the structure into “SVA1.1.1”Intrinsic value”, it contains five measurable VCs: “VC1.1.1.1: Functionality”, “VC1.1.1.2: Reliability”, “VC1.1.1.3: Usability”, “VC1.1.1.4: Maintainability” and “VC1.1.1.5: Portability”. The criteria used for measuring each of the VCs are given in the “Measured through” attribute of the corresponding VC. The entire SVM together with detailed and complete attributes can be downloaded [40].
5.3 Interrelationships

Interrelationships in the value map are possible and common among value aspects. We do not claim to map all possible relationships, the rationale being SVM is an evolving entity and so are the interrelationships. The interrelationships detailed thus far in SVM were identified using three sources: 1) interrelationships that were stated in the selected papers, 2) interrelationships that were obvious during categorization and placement of VAs/SVAs and VCs, and 3) interrelationships that were identified during the creation of patterns in the industry case study (see Table 6). For example, “SVA: PVq-product architecture value”, from the internal business perspective is related to the external quality attributes of a product’s perceived importance from the customer perspective “SVA: intrinsic value” [P44]. This is due to that if the software architecture is not designed with flexibility, reliability, usability, maintainability and portability in mind, it will not be possible to provide users/customers with software that is in turn flexible, reliable, usable, maintainable and portable. Moreover, when determining differentiation with respect to perceived value for customer “SVA: Differentiation wrt. perceived value”, it is related to the sub-value aspects of perceived value from customer’s perspective “VA: perceived value”. This is particularly important when companies decide differential advantages of their software products as they would strategically plan to differentiate on value aspects that are valued more by their customers.

Looking into the innovation and learning perspective, “VA: value of market” is related to “VA: value of technology”. For example, duration of income in “VA: value of market” is dependent on the life of subject technology “VA: value of technology” and amount of income is influenced by proprietary position or contribution ratio of subject technology [P67].
Thus, if interrelated value aspects are considered independently, a number of VCs may be overlapping or redundant and the accuracy of valuation may deteriorate. One of the contributions of the Software Value Map is to make these interrelations explicit to avoid blind addition of the overlapping and redundant value components. However, this has not been addressed during the industrial evaluation of SVM, and is an area of future research during evaluation and evolution of the SVM.

Table 6 – Interrelationships within Software Value Map

<table>
<thead>
<tr>
<th>VA/SVA</th>
<th>VA/SVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP3: Internal business perspective</td>
<td>VP1: Customer perspective</td>
</tr>
<tr>
<td>SVA3.1.2.3.3: PVq-product architecture value</td>
<td>SVA1.1.1: intrinsic value</td>
</tr>
<tr>
<td>VP3: Internal business perspective</td>
<td>VP1: Customer perspective</td>
</tr>
<tr>
<td>SVA3.2.1: Differentiation wrt. perceived value</td>
<td>VA1.1: perceived value</td>
</tr>
<tr>
<td>VP4: Innovation perspective</td>
<td>VP4: Innovation perspective</td>
</tr>
<tr>
<td>VA4.2: value of technology</td>
<td>VA4.2: innovation value for market</td>
</tr>
<tr>
<td>VP1: Customer perspective</td>
<td>VP1: Customer perspective</td>
</tr>
<tr>
<td>SVA1.1.5.1: Pragmatic value</td>
<td>SVA1.1.1: intrinsic value</td>
</tr>
</tbody>
</table>

6. Step 4: INDUSTRIAL EVALUATION THROUGH APPLICATION OF SVM

The Software Value Map in itself offers a large overview of the concept of value from multiple perspectives, although useful for many tasks, practitioners in a company generally work with one task at a time. For example, a project manager may struggle with the decision whether or not to put extra resources into the development of a certain component in order to make it easily reusable, and a product manager may struggle with the decision relating to requests for a product customizations. In the case of the project manager, he needs to evaluate the potential value of making e.g. a feature or function reusable. In the case of the product manager, there has to be an evaluation of the potential value (or lack thereof) of the customization and the ability to weigh customization requests against each other.

In these and other specific cases using the complete value map is impractical, as traversing the entire map would imply going through all the value components. For this reason, the concept of Impact evaluation pattern was introduced, consisting of a sub-selection of value components. Each pattern is adapted for a specific task, directly supporting the practitioner. These Impact evaluation patterns were created using the data collected in the evaluation study (Step 4, see Figure 2) conducted at Ericsson, our main industry partner for this project.

Ericsson is a world leading telecommunication system provider, offering 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. Within Ericsson, like other software product development companies, they are faced with the challenge of making decisions incorporating value aspects and value components relevant from different perspectives rather than taking product development decisions based only on short terms costs and immediate sales.

The idea behind Impact evaluation patterns is described in Section 6.1, and Section 6.2 details the steps for pattern creation. In total four patterns were created at Ericsson, for four different roles involved in a particular decision-making scenario (fully described in detail online [40]). Section 6.3 describes in detail the how the created patterns for decision-support can be and
were used. Section 6.4 concludes the evaluation section by presenting lessons learned from the use of the Software Value Map and the creation of patterns.

6.1 The Concept of Impact Evaluation Patterns

The concept of Impact evaluation patterns was inspired by software design patterns. In software engineering, a design pattern is a general reusable solution to a commonly occurring problem in software design [55]. The same philosophy was used to identify Impact evaluation patterns in different decision-making scenarios. An Impact evaluation pattern can be described as a generally reusable solution for a commonly occurring decision-making challenge in a particular scenario. For example, a product manager can use an Impact evaluation pattern for initial screening to decide if a set of new requirements should be selected for implementation in the product or not. For large companies, like Ericsson, who have multiple products, and a large number of people working in the same role, common evaluation criteria for a particular decision is fundamental for homogenous decision-making, which is achieved using a pattern.

Each pattern is documented/specified through a number of attributes, as can be seen in Table 7. Here we can see that attribute 5: “Value aspects” would contain VCs that are used in the pattern, for example “VC1.1.1.1: Functionality”, “VC1.1.1.2: Reliability” and “VC.1.1.2.1: Deliver process value wrt. time” can be included in a requirement selection decision-making pattern. Attribute 6: “Impact evaluation criteria” would contain rubrics-based criteria to evaluate the impact on value aspects given in attribute 5.

<table>
<thead>
<tr>
<th>Table 7 - Impact evaluation pattern attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. <strong>Pattern Name and Classification</strong>: A descriptive and unique name that helps in identifying and referring to the pattern.</td>
</tr>
<tr>
<td>2. <strong>Intent and Motivation</strong>: A description of the goal (decision-making challenge that is faced) behind the pattern and the reason for using it.</td>
</tr>
<tr>
<td>3. <strong>Also Known As</strong>: Other names for the pattern.</td>
</tr>
<tr>
<td>4. <strong>Applicability</strong>: Situations in which this pattern is usable; the context for the pattern.</td>
</tr>
<tr>
<td>5. <strong>Value aspects</strong>: A hierarchical branching of the VPs, VAs, SVAs and VCs used in the pattern.</td>
</tr>
<tr>
<td>6. <strong>Impact evaluation criteria</strong>: A rubrics-based criteria to evaluate the impact on value aspects given in the pattern.</td>
</tr>
<tr>
<td>7. <strong>Relationships</strong>: If SVAs or VCs have interrelationships a description of what the interrelationships are and how they affect the overall pattern.</td>
</tr>
<tr>
<td>8. <strong>Consequences</strong>: A description of the results (valuation), side effects, and tradeoffs caused by using the pattern.</td>
</tr>
<tr>
<td>9. <strong>Related Patterns</strong>: Other patterns that have some relationship with the pattern; discussion of the differences between the pattern and similar patterns.</td>
</tr>
<tr>
<td>10. <strong>Involved stakeholders</strong>:</td>
</tr>
<tr>
<td>• A list of stakeholders involved in creating the pattern.</td>
</tr>
<tr>
<td>• A list of stakeholders involved in verification of the pattern.</td>
</tr>
<tr>
<td>• A list of stakeholders actually using the patterns.</td>
</tr>
<tr>
<td>• A list of stakeholders maintaining the pattern.</td>
</tr>
<tr>
<td>• A list of stakeholders providing measurements/assessments for the value components included in the patterns.</td>
</tr>
</tbody>
</table>

Since an Impact evaluation pattern would be a living entity it is necessary to keep a record of all the stakeholders involved to ensure that pattern creation, verification, usage and maintenance responsibilities are explicitly stated and in case of modifications required in the pattern, relevant stakeholders can be contacted.
6.2  Step 4.1: Pattern creation process

A total of four patterns were created for four different roles as a part of the industrial evaluation case study at Ericsson. This section goes through the steps involved in creation of Impact evaluation patterns (Step 4.1, see Figure 1).

6.2.1  Step 4.1.1: Scenario selection

The first step in evaluation was to identify a scenario. This could be any scenario that has different decision-making activities related to the evaluation of software creation, updating, customization, and/or modification. A scenario can have one or more discrete decision-making activities. As a result of a joint discussion with the industry practitioners one major scenario was chosen, namely product customization (PC). The rationale being that Ericsson has many PCs in their day-to-day work, and the potential optimization and introduction of a rich view in relation to evaluation of PCs was recognized as potentially very beneficial.

A product customization can be defined as a modification of the source code of an Ericsson product, initiated by a customer request. A product customization can consist of new functionality in a product that is not planned for in the product roadmap, or new functionality in a product that is planned for in the product roadmap, but not in the timeframe as requested by the customer. In case of product customizations (PCs) the actual development costs will have to be paid by the customer, and fast delivery is central. However, the potential value impact of a PC on the present product implied by a customization is not always analyzed, as the estimation of “value” was often limited to cost and estimated revenue. To achieve a deeper value impact analysis, an Impact evaluation pattern creation was initiated.

Within product customization there can be different decision-making activities at four different stages, as identified in relation to how Ericsson optimally wanted to work:

1. **Initial screening:** this involves a quick value analysis, should the PC in question be realized or dismissed
2. **Pre-analysis:** this requires a relatively detailed value analysis from business and technical perspectives
3. **Enable reuse:** this activity requires performing value analysis in order to identify which PC as such can be made reusable or what parts of the PC should be made reusable to enable faster handling of future PCs and faster delivery of product releases. The main decision being should one spend extra resources to develop the PC in a reusable fashion or should it be done with time-to-market as a focus
4. **Post delivery analysis:** involves performing value analysis to measure if the value predicted for a PC in the “Initial screening” stage is gained or not. This analysis is done after the PC has been implemented and delivered to the customer/market

Within the PC scenario, one of the listed decision-making activities was selected to be studied, and included in the pattern, namely pre-analysis. The pattern created for pre-analysis used by a system manager for impact evaluation is used as an example in this paper.

6.2.2  Step 4.1.2: Role Identification

The next step, after the selection of a scenario and corresponding decision-making activity, was to identify the roles involved in that decision-making activity. These practitioners would be the main contributors in the creation of the pattern. Members of the strategic product management organization at Ericsson supported in the identification of the
roles and participants for the study.

Table 8 contains the roles directly involved in pre-analysis decision-making activity. At least two practitioners in each role were chosen to participate in the evaluation study.

Table 8 - Roles and participants in the study

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>Number of participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software product manager (SPM)</td>
<td>A software product managers deals with product release, product market and general functionality.</td>
<td>2</td>
</tr>
<tr>
<td>System manager (SM)</td>
<td>A system manager is responsible for the technical aspects of a software product.</td>
<td>2</td>
</tr>
<tr>
<td>Solutions architects (SA)</td>
<td>A solution architect is responsible to design a solution given a set of requirements from the customer. The solution proposed may be then evaluated by system managers.</td>
<td>4</td>
</tr>
<tr>
<td>Customer unit representatives (CU)</td>
<td>A CU representative is responsible for the business case of a set of requirements. He/she is also responsible for the pricing of the solutions.</td>
<td>2</td>
</tr>
</tbody>
</table>

After the participants were selected, a three-hour work session was held with each participant. A work session started with a 15-20 minutes introduction followed by Step 4.1.3 and Step 4.1.4, described below. The work sessions were performed in an interactive manner, discussing and elaborating on what aspects were presently considered critical for the decision-making activity, as well as what additional analysis/value components could be beneficial and should be added to the pattern.

6.2.3 Step 4.1.3: Eliciting current decision-making criteria

In order to create the Impact evaluation pattern, it was important to first elicit what value aspects were considered at present by the participants when deciding if a PC should be realized or not (prior to the introduction of the value map and Impact evaluation pattern). This activity was important to “benchmark” the current decision criteria. The Software Value Map was not shown to the participants prior to this step, as we did not want to influence their answers. In total the following information was elicited from the participants:

1. Value aspects: Value considerations that the decision maker has in a particular decision-making scenario
2. Description: In order not to misinterpret a value aspect it was necessary to record description of every value aspect as stated by the participant
3. Measured today: Against each value aspect, the participant was asked how the stated value aspect is measured/evaluated
4. Should be measured: In order to identify if measurement/evaluation of the stated value aspect needs to be more refined or coarse, the participants were asked how it should ideally be evaluated/measured.

Table 9 gives an example of the type of answers collected from the participants.
As can be seen in Table 9, system managers described cost, impact on different nodes, and architectural value among the criteria used for taking decision about a PC in the pre-analysis phase. However, for certain value aspects, like architectural value, they could not state which explicit architectural value aspects were evaluated. They were not satisfied with the subjective evaluations; however it was difficult for them to suggest ideal measures for the value aspects they stated. The fundamental difficulty was to identify a common evaluation scale for all the value aspects.

### 6.2.4 Step 4.1.3: Eliciting ideal value aspects and components

After benchmarking the current decision criteria, all relevant value aspects and components were elicited from the participants, in essence trying to catch the ideal aspects. In this step, the value map was shown to the participants, and all the VCs in the value map were presented to the participants in a form. An extract of this form can be seen in Table 10 (the complete form is available in the online manual [40].

<table>
<thead>
<tr>
<th>Value Components</th>
<th>Perspective and description</th>
<th>Applicability</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer perspective</td>
<td>The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.</td>
<td>Fully applicable = 3, Applicable = 2, Kind of applicable = 1, Not sure = 0, Not applicable = -1</td>
<td></td>
</tr>
<tr>
<td>Functionality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>The capability of the software product to maintain a specified level of performance when used</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For each value component, “perspective and description” field contained one of the four perspectives: customer, internal business, financial or innovation and learning. The description was copied from the “Description” attribute of the VC in the SVM. The participants were asked to indicate the applicability of a particular VC in the specific decision-making activity. The choices ranged from fully applicable to not applicable. Further, since one of the objectives of the case study was to elicit which value components participants wanted to consider ideally, it was required to elicit how coarse or refined they wanted a particular value component to be measured/evaluated. This information was elicited to make the Impact evaluation patterns usable and useful.

6.2.5 Step 4.1.5: Data analysis and creation of Impact evaluation pattern
Conducting Step 4.1.3 and Step 4.1.4 the work sessions indicated that currently decision-making was cost and immediate revenue centric. While system managers implicitly consider effects on the architectural value of a product, they do not have any explicit measurements/evaluations of the architectural value when deciding if a PC should be done or not. Further, system managers mentioned that because they do not have architecture value evaluations as a specific value aspect in their work at present, it was almost impossible to show higher management that the PC in question, although promising higher short-term revenues, would have a devastating effect on the product’s value in the long run.

Figure 5 – Pre-analysis impact evaluation pattern for system managers with Critical path
In order to create an Impact evaluation pattern for a role (SM in this illustrated case), we set up rules for identifying what value components should be included in the pattern (based on the questionnaire filled by participants in Step 4.14). These rules are given below:

- **Rule 1**: For all the participants in one role (for example System management) take average of the values assigned to every VC. If average is greater than or equal to 2, it is included in the pattern

- **Rule 2**: If the difference between the values assigned by the participants (difference of opinion) is greater than or equal to 2, more analysis is required.

Figure 5 shows the value components selected by system managers that should be considered for analyzing a PC in question. Starting from the left, system managers believed that during pre-analysis they should evaluate functionality value from the customer perceived value perspective.

From the internal business value perspective, system managers were of the opinion that while analyzing a PC, competitive market requirements value from the resell perspective (“VC3.1.1: Market requirements value”) should be evaluated. This is certainly relevant from the internal business value perspective as if the PC in question gives competitive advantage, and/or can be resold, it would add to the value for the business. System managers also selected some architectural VCs (“VC 3.1.2.3.3.1: Functionality”, “VC 3.1.2.3.3.1.3: Usability”, “VC3.1.2.3.3.1.4: Maintainability” and “VC 3.1.2.3.3.1.5: Portability”) to be evaluated for explicit analysis of how architecture of the product is affected by implementing the PC in question. It could be either improving architecture value or degrading it. And last but not the least, project value with respect to time and cost should be evaluated.

The attributes of the created Impact evaluation pattern are given in Table 11.

<table>
<thead>
<tr>
<th>1. Pattern Name and Classification</th>
<th>Pre-analysis Impact evaluation pattern for SM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Intent and Motivation</td>
<td>To perform a detailed value analysis from the business and technical perspectives This was stated as the main motivation by Ericsson practitioners to have pre-analysis Impact evaluation pattern.</td>
</tr>
<tr>
<td>3. Also Known As</td>
<td></td>
</tr>
<tr>
<td>4. Applicability</td>
<td>to analyze and decide if a PC in question should be done or not</td>
</tr>
<tr>
<td>5. Value aspects</td>
<td>see Figure 5</td>
</tr>
<tr>
<td>6. Impact evaluation criteria</td>
<td>see Table 12</td>
</tr>
<tr>
<td>7. Relationships</td>
<td>Business value contains Value for Customer and Product Value</td>
</tr>
<tr>
<td>8. Consequences</td>
<td>//to be added when the pattern is put into actual use</td>
</tr>
<tr>
<td>9. Related Patterns</td>
<td></td>
</tr>
<tr>
<td>10. Involved stakeholders</td>
<td>SMs, SPMs and marketing department</td>
</tr>
</tbody>
</table>

For some VCs it was easy to put measurement/metrics parts (for example VC3.1.2: Physical value wrt. cost), where monetary measures can be used. For others it was hard to estimate impact in e.g. monetary terms (for example, VC1.1.1.1: Functionality value), therefore it was agreed to use rubrics-based expert opinion criteria for Impact evaluation. The rubrics designed for Impact evaluation of VCs given in the system managers’ pattern are given in Table 12 (Appendix A).

From Figure 5, it can be seen that an Impact evaluation pattern for pre-analysis for SM resulted in having seven VCs to be evaluated for each decision. For SPMs, the VCs in the pattern went up to 13. During the work sessions while the participants realized that a lot of VCs are relevant and should be considered in their decision-making, they raised concern as to the practicality of using patterns that were large (many VCs). This was a correct
observation, and motivated the creation of the concept of critical path and value analysis triage (Step 4.1.6 in Figure 1), detailed below.

6.2.6 Step 4.1.6: Value Analysis Triage and Critical Path

In market driven software product management and development, optimization of resources expenditure for decision-making is critical [7, 30, 31, 56-58]. This demands scalable and efficient decision support methods/models/tools to be developed.

The Software Value Map (large) was tailored and a subset of VCs was selected to create a pattern (smaller), fitting a specific decision-making scenario and role. As the next step to even further optimize the potential use of the value map, critical path was introduced as a concept. The critical path is a “deal breaker” part of a pattern that is one or several VCs that can be evaluated first when using a pattern. If these do not produce satisfactory results, the rest of the pattern can be ignored, as the decision is “No” in any case. This triage of value aspects was inspired by [59] [60] and their application of the concept of requirements selection. We do the same using critical paths, but apply it on what VCs to look at first when using a pattern, to, for example take a decision whether or not to accept a product customization for a product.

In order to identify the VCs included in the critical path, another rule was introduced (as elicited in Table 10).

- **Rule 3:** For all the participants in one role (for example system management) take average values assigned to every VC. If average is greater than or equal to 2.5, the corresponding VC is a critical path candidate.

In the example used here, the system managers considered “VC3.1.2.3.3.1: Functionality” to be in their critical path (see Figure 5, the VC on critical path is made bold and has larger font size). System managers belong to the development unit. Their central responsibility is to sustain the existing functional design of the architecture, thus functionality value from the product’s architecture value perspective (“VC3.1.2.3.3.1: Functionality”) constitutes critical path in their Impact evaluation pattern.

There can be more than one VC in a critical path as can be seen in the patterns for SPMs, SAs and CU representatives [40]. During the creation of Impact evaluation patterns using SVM, identifying critical paths for each Impact evaluation pattern emerged as a fundamental activity.

6.3 Step 4.2: Using the Patterns

Once the patterns and critical paths were created, an evaluation of their usability and
usefulness was deemed a natural next step. A simple excel based prototype tool called the Software Value Analysis tool (see Figure 6) was created to aid the practitioners when using their patterns. As an example Figure 6 shows the system managers pre-analysis Impact evaluation pattern. For the examples here product customization requests (PCs) are used, however, the use of patterns can be on any artifact or phenomenon (e.g. requirements). While evaluating a PC request against VCs in the pattern, the system manager simply has to choose one out of five options given in relation to each VC. For example, for the VC “Market requirements value”, the options are:

+2: More than $n$ additional operators might buy the product with this functionality
+1: More than $m$ additional operators might buy the product with this functionality
0: No additional customers will buy this functionality
-1: Operators will be hesitant to buy this functionality
-2: Operators will not buy this functionality

A system manager has the possibility to either evaluate the PC against entire pattern or use the critical path VCs first for quick and dirty analysis. Once a PC has been evaluated against the patterns, several interesting visual aids can be employed, aiding discussions and decision meetings. Examples of such graphics are detailed below.

6.3.1 Theme based Analysis

In addition to patterns, Themes were developed. Themes are effectively a weighting of value aspects depending on the current focus of the development organization. A theme can be anything that is mandated by the product strategies. For example, for one year/period a theme might be to focus on ARCHITECTURE or maybe on INNOVATION. In case of the innovation theme the value aspects such as "differentiation value" and "Innovation value of technology" (nested under that area in the SVM) are prioritized over other value aspects such as "Architectural usability value" (and the other way around if using the architecture theme). By changing themes (one click in the tool) one can compare a specific PCs value in relation to a theme (which also can be seen as a benchmark to accept or reject a PC) not only based on the estimation of the value aspects, but also depending on timing and strategies.

Figure 7, Figure 8 and Figure 9 illustrate the use of patterns and the evaluation of a specific product customization. On the horizontal axis all VCs in the pre-analysis Impact evaluation pattern for system managers and VCs in the current theme are listed. On the vertical axis whole numbers showing value of VCs are listed. If the current theme is the “architectural theme”, then according to a system manager’s evaluation, the product customization CIN-191 is perfectly aligned with the existing focus (see Figure 7) and can be accepted for implementation without any issues. However, if the theme is changed from “Architectural theme” to “Innovation theme”, Figure 8 clearly shows that CIN-191 is not the best choice for implementation, as it does not add positively to the value components prioritized with the “Innovation theme”.

Page 151 of 231
This one-click theme based analysis was very appreciated by the practitioners during the evaluation of the patterns and tool. Similarly, such an analysis can be done simultaneously for multiple PCs to select the ones more aligned with the current theme. For example, Figure 9 shows CIN-191 and a new product customization MOBI-056. Both are well aligned with the current focus and positively impact the architectural value of the product, however, CIN-186 could, if implemented, decrease the maintainability value of the architecture. Through this tool practitioners can quickly benchmark PCs against each other, and against the current theme.
6.3.2 Internal versus external value analysis

Another interesting analysis possibility provided through the use of patterns is the placement of e.g. PCs in relation to their impact on internal and external value. All the VCs in SVM are classified as contributors to either external or internal value. Figure 10 shows four quadrants in which PCs can be placed. The top-right quadrant is the one where PCs positively impacting both external value (for the customer), and internal value (for the development company) are placed. The top right quadrant can be seen as a win-win situation both for the customers and Ericsson in this case. Whereas, the top-left quadrant would contain PCs that positively impact/add to the internal value however, negatively affect external value. For example, a PC might improve the architecture’s maintainability while simultaneously negatively affecting the efficiency of use of the product. And if the cumulative internal value (after summing values for all VCs contributing to internal value) is positive and cumulative external value is negative (after summing values for all VCs contributing to external value), the PC gets placed in this quadrant. For PCs in this quadrant, Ericsson needs to carefully analyze the gains in internal value with respect to losses in external value before accepting the PCs. The bottom right quadrant contains PCs that have higher external value however, negative internal value. In such cases, costs for avoiding/minimizing the negative impact on internal value need to be calculated and added to the cost of the PC. Such PCs can still be accepted if the customer is willing to pay the additional cost of the negative internal impact. The PCs lying in the bottom left quadrant are candidates for rejection as they neither add to external nor to internal value.

During industry evaluation of the patterns and tool, practitioners pointed out that the placement of PCs in relation to external value and internal value can be used in retrospective also, for analyzing the quality of decisions post-fact. The decisions made using pre-analysis Impact evaluation patterns can be recorded and once every six months a sample of PCs can be picked and the pre-analysis Impact evaluation pattern for each one of them is filled out again, based on the post-delivery facts (how were the internal and external values actually impacted), and compared to the estimates. For example, Figure 11 shows that for CIN-191 estimated external value and internal value impacts (20, 46) were more than actually achieved (10, 20). Similarly, estimated internal value projection (27) for MOBI-056 was not actually achieved (15). On the other hand, the positive aspect is that actual negative impact on external value (-5) was less than the estimated one (-10). Keeping a record of decisions
made based on Impact evaluation patterns can help practitioners learn from their estimations and decisions and consequently improve the quality of both.

Figure 10 – Analysis of PCs in relation to external versus internal value
6.3.3 Role-based Analysis

In addition to looking at comparing PCs, and PCs and themes, a comparison between roles can be performed. The central aim of this is to explicitly compare different perspectives and to be able to lift a PC’s different impact on the product. In Figure 12, on the horizontal axis one can see all VCs in the pre-analysis Impact evaluation pattern for system managers, and the VCs in the current theme are listed. On the vertical axis whole numbers showing value of VCs are listed. The bars represent the ideal values according to the selected theme, Impact evaluation done by software product managers, and system managers according to their corresponding rubrics. From Figure 12 it is interesting to note that a product manager deems the architectural usability impacted as POSITIVELY, while the system manager deems the SAME value aspect as being NEGATIVELY impacted. By catching these types of “problems” and inconsistencies, explicit discussions can result in well-founded decisions without escalations due to implicit knowledge and judgment. Similar analysis can be done involving all the stakeholders involved in decision-making.
6.4 Results and Lessons Learned

6.4.1 Challenges addressed by SVM and Impact evaluation patterns

This section describes the lessons learned in relation to the use of SVM and patterns — and relates it to the challenges described in Table 1 (see Section 2).

Communication enabler

One of the major contributions of SVM is that it offers a unified view of value, which can be used by professionals to develop a common understanding and vocabulary pertaining to value, as well as acting as decision support to assure no value perspective is unintentionally overlooked when taking product management and development decisions (addressing Challenge C1, see Table 1). This made communication between e.g. a solution architect, a product manager, system manager and a customer unit representative not only possible (addressing Challenge C1), but explicit as SVM was used as a base for the communication and the impact of the decision could be weighed (addressing Challenge C4). Prior to SVM, communication was often based on an assumed definition of value, which varied between stakeholders. These variations were often not caught; rather the consequences of them were seen in failures and problems down the line.
Value-based requirements selection

In order to address C2: Long-term versus short-term gains and losses and C4: Value-based impact and consequence analysis (see Table 1), SVM can be used as a consolidated view of value including value components required to be evaluated during requirements selection; balancing short-term and long-term gains and sacrifices keeping in view the entire product life cycle. For example, during requirements selection in addition to short term increases in customer value and company revenue, a company’s and product’s long-term sustainability view can also considered. For example evaluating the effects of a requirement on the maintainability value of the product’s architecture (VC 3.2.1.1.1.4, see Figure 4), human capital value (VC 4.1.1, see Figure 4) of the company and innovation value for the market (VA 4.3, see Figure 4) would achieve a more comprehensive (long-term) impact analysis of a certain decision. The overall trade-off between positive and negative impact on the present product offering can be estimated. This is central from many perspectives. For example, from a business perspective, the selection and realization of a feature might be good idea, but simultaneously the long-term effects pertaining to e.g. system architecture, might be very negative. Another less obvious example might be that the functionality value for one customer, and thus revenue, might be high when implementing Feature X, but the effect for other customers might be negative in relation to the inclusion of said feature, e.g. decreasing interoperability value (part of VC1.1.1.1:Functionality, see [40]).

Reduced time-to-decision

SVM was used as a basis to derive different Impact evaluation patterns which can reduce time-to-decisions in relation to a wide range of software product development situations, like reuse and customer tailoring (for details see Section 6). An Impact evaluation pattern can be described as a general reusable solution for a commonly occurring decision-making challenge in a particular scenario. The concept of “critical path” was also introduced and evaluated to further reduce the time-to-decision by identifying the value components considered to be “deal-breakers” within each pattern (for details see Section 6.2.6).

Retrospective analysis

If Impact evaluation patterns are used for decision-making, a common value-based set of criterion is used by all the practitioners. By recording the decisions made using the patterns, for example the requirements selection pattern (the decision is documented according to the pattern’s value components); a retrospective examination of release planning decision-making can be made, at a time when the consequences of requirements selection decisions are visible (thus addressing challenge C3: release planning quality). And, by analyzing the decision outcome in retrospect, organizations can gain valuable knowledge of how to improve the requirements selection process and increase the chances of market success.

In summary, in addition to the stated points above, the industry practitioners at Ericsson participating in the creation and evaluation of patterns gave the following feedback in relation to the SVM and the use of Impact evaluation patterns:

a. SVM was viewed as a dictionary that offers a modular view, and common definitions of software value aspects, sub-value aspects and value components

b. It was possible to identify value components that might be important to consider in a certain scenario and for a particular pattern. Even if this was not an exact science, and many times too many value components were selected (as they could be relevant), the possibility of a more complete view beyond the obvious was welcomed. The challenge was to iteratively refine the patterns to make them complete enough but at the same time
not too large to be usable.

c. The use of critical paths in Impact evaluation patterns was a good way in which fast pre-screening could be performed prior to filling in the complete pattern. The implicit challenge here was that if a practitioner only fills in the (few) value components in the critical path, and others fill in the entire pattern, comparisons (as shown in Figure 12) would be harder as information was missing.

d. The Excel tool for software value analysis was highly appreciated with respect to usability and usefulness [61].

As SVM was introduced the complexity of the value concept also increased. This can for example be seen in the fact that some VCs can be an aggregate of two or more value components, for example, “VC1.1.4: Complementary value” could be further categorized as “VC: Complementary value for a product” and “VC: Complementary value for a service”. This example is relevant when taking a decision regarding a product customization, given if the PC in question increases complementary value for another product or another service. In addition, as noticed by practitioners at Ericsson, the use of SVM and the patterns can be seen as increasing the overall effort put on the analysis of a artifact or phenomenon (e.g. a PC or requirement). Using a pattern might imply filling in 5-10 VCs. However, the explicit nature of the patterns, and the possibility to be explicit and use a shared vocabulary inherent in the use of the patterns, analysis time was saved overall.

7. CONCLUSION

The Software Value Map presented in this paper contributes as a first step towards a complete categorization of value aspects, and the division into sub-value aspects and measurable value components. This was done utilizing knowledge from state-of-the-art in software engineering, business, management, and economics, gathered through extensive literature reviews, but also working with professionals in industry.

Moving away from a cost-based perspective, to a value-based perspective, utilizing both well-known aspects, such as customer satisfaction, but at the same time introducing more complex and complete aspects such as the customers’ perception of value, is central. In themselves, most constituents of the value map are not new, but collected from different disciplines. The contribution of the Software Value Map is the collection, categorization, and qualification of aspects adhering to the original intent, but also adding the concepts of interdependencies, attributes and measurement (where available), as well as the angling of value perspectives towards impact evaluation. In addition, the industrial evaluation of the value map was preceded by the identification of scenarios and subsequent creation of Impact evaluation patterns. Impact evaluation patterns are purpose build incarnations of the complete value map, usable and useful for specific tasks. In addition to this, to further enhance usability and reduce time-to-decision, the concept of critical paths was introduced to enable triage of decisions. Thus transforming a complete value map to a usable and useful tool was also of paramount importance.

A fundamental impact within Ericsson, attributed to the creation and use of the Software Value Map, is the shift from cost-based discussions and reasoning, to value-based decision support. Even without the creation of patterns the value map gave a common vocabulary to the professionals, enabling them to be precise in their discussions of value, giving them the possibility to confirm exactly what they all mean by “value”. The industry practitioners did not only verify the benefits of having a consolidated view of value components, relevant for a particular Impact evaluation pattern, for decision-making; they also found the Software Value Map a step towards common definitions and understanding of value components enabling effective communication. Additionally they viewed the value map as a basis for
customer value analysis, improving their understanding of what constitutes value for their customers, as well as for themselves. And last, but not the least, by recording the decisions taken by evaluating value components in the corresponding Impact evaluation patterns, a retrospective analysis of the decisions for improving requirements selection quality was also seen as a major contribution by the practitioners.

From a research perspective, researchers can use the Software Value Map as a collection of different value aspects and value components that need to be evaluated and compared for taking decisions about software management and development. This would enable devising measurement/evaluation solutions that do not just evaluate one aspect of value but rather evaluate all the different aspects spanning from customer perceived value, to internal business value and architectural impact and reusability. The idea is for researchers to use SVM as a starting point, where additions and refinements can be made, and any number of patterns can be created and evaluated in industry. The key is that if a pattern is incomplete, and the addition needed cannot be found in the SVM, then an addition can and should be made in the SVM.

“Completeness” was never the goal of the SVM, rather to create an as complete view on value as possible. The SVM should be used and extended, complementing value aspects and perspectives as needed, but maybe more importantly, we need to improve upon how these aspects should be measured in industry - a central avenue for future research in value based software engineering.

8. Acknowledgements

We would like to thank our Ericsson steering group members and all the participants of the study who provided valuable input during the creation and validation of SVM and Impact evaluation patterns. We would further like to extend our thanks to the reviewers from Ericsson for reviewing the paper and providing valuable feedback which has helped in approving the paper.

9. REFERENCES


### Appendix A

#### Table 12 – VCs with corresponding rubrics for Impact evaluation

<table>
<thead>
<tr>
<th>VC1.1.1.1: Functionality value</th>
<th>Impact</th>
<th>Side effects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+2</td>
<td>critical</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Important</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>No effect</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>Wrongly specified</td>
</tr>
<tr>
<td></td>
<td>-2</td>
<td>Completely misunderstood</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Impact</th>
<th>Side effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>Supports standard features</td>
<td></td>
</tr>
<tr>
<td>+1</td>
<td>Supports optional feature</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>No effect</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>Blocks/removes optional feature</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>Blocks/removes standard feature</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VC3.1.1: Market requirements value</th>
<th>Resell/standard product</th>
<th>Effort required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+2</td>
<td>Minimal effort required to make it resellable or add to standard product</td>
</tr>
<tr>
<td></td>
<td>+1</td>
<td>Some effort required to make it resellable or add to standard product</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>-1</td>
<td>Customers will be hesitant to buy product with this feature</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>Customers will not buy product with this feature</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VC3.1.2.1: Physical value wrt. cost</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>Minimal cost to implement this feature</td>
</tr>
<tr>
<td>+1</td>
<td>Some cost to implement this feature</td>
</tr>
<tr>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>-1</td>
<td>Substantial cost to implement this feature</td>
</tr>
<tr>
<td>-2</td>
<td>Huge cost to implement this feature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VC3.1.2.3.3.1: Architectural functionality</th>
<th>+2</th>
<th>Greatly improves the existing functional design of the architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+1</td>
<td>Improves the functional design of the architecture</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>No effect</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>Degrades the existing functional design of the architecture</td>
</tr>
<tr>
<td></td>
<td>-2</td>
<td>Seriously degrades the functional design of the architecture</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VC3.1.2.3.3.3: Architectural Usability</th>
<th>Understandability</th>
<th>Learnability</th>
<th>Operability</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>Greatly improves understandability of the architecture</td>
<td>Greatly improves learnability of the architecture</td>
<td>Greatly improves operability of the architecture</td>
</tr>
<tr>
<td>+1</td>
<td>Improves understandability of the architecture</td>
<td>Improves learnability of the architecture</td>
<td>Improves operability of the architecture</td>
</tr>
<tr>
<td>0</td>
<td>No affect</td>
<td>0</td>
<td>No affect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>No affect</td>
</tr>
</tbody>
</table>
### VC3.1.2.3.3.4: Architectural maintainability

<table>
<thead>
<tr>
<th>Analysability</th>
<th>Changeability</th>
<th>Stability</th>
<th>Testability</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
</tr>
<tr>
<td>Greatly improves</td>
<td>Greatly improves</td>
<td>Greatly improves</td>
<td>Greatly improves</td>
</tr>
<tr>
<td>analysability of the</td>
<td>changeability of</td>
<td>stability of the</td>
<td>testability of the</td>
</tr>
<tr>
<td>architecture</td>
<td>the architecture</td>
<td>architecture</td>
<td>architecture</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>Improves</td>
<td>Improves</td>
<td>Improves</td>
<td>Improves</td>
</tr>
<tr>
<td>analysability of the</td>
<td>changeability of</td>
<td>stability of the</td>
<td>testability of the</td>
</tr>
<tr>
<td>architecture</td>
<td>the architecture</td>
<td>architecture</td>
<td>architecture</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>No affect</td>
<td>No affect</td>
<td>No affect</td>
<td>No affect</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Degrades</td>
<td>Degrades</td>
<td>Degrades</td>
<td>Degrades</td>
</tr>
<tr>
<td>analysability of the</td>
<td>changeability of</td>
<td>stability of the</td>
<td>testability of the</td>
</tr>
<tr>
<td>architecture</td>
<td>the architecture</td>
<td>architecture</td>
<td>architecture</td>
</tr>
<tr>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>Seriously degrades</td>
<td>Seriously degrades</td>
<td>Seriously degrades</td>
<td>Seriously degrades</td>
</tr>
<tr>
<td>analysability of the</td>
<td>changeability of</td>
<td>stability of the</td>
<td>testability of the</td>
</tr>
<tr>
<td>architecture</td>
<td>the architecture</td>
<td>architecture</td>
<td>architecture</td>
</tr>
</tbody>
</table>

### VC3.1.2.3.3.5: Portability

<table>
<thead>
<tr>
<th>Portability</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
</tr>
<tr>
<td>+1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>-1</td>
</tr>
<tr>
<td>-2</td>
</tr>
</tbody>
</table>
ABSTRACT

Context: Value Stream Mapping is a method of Lean product development and production, which has recently attracted interest in the software engineering community. In other contexts (such as military, health, production), Value Stream Mapping has achieved considerable improvements in processes and products. The goal is to also leverage on these benefits in the software engineering context.

Contribution: In the product development domain a manual has been provided of how to conduct Value Stream Mapping. The researchers have extended this manual and created a framework outlining different strategies for conducting Value Stream Mapping in the software engineering context. The framework is a detailed description that can serve as a manual for practitioners who seek to adopt Value Stream Mapping in their organizations.

Method: A case study has been conducted instantiating the framework at Ericsson AB. The case study results consist of two parts. First, the instantiation and motivations for selecting certain strategies have been provided. Second, the outcome of the value stream map activity is described in detail.

Results: The results showed that valuable insights can be obtained in a very limited time frame (10 hours with 12 persons). The richness of the results is visible by the number of improvement opportunities identified across the overall development life cycle, as well as the improvement suggestions and reflections regarding the new process definition implementing the improvements. In addition, multiple value considerations relevant from different perspectives (customer value, internal efficiency, delivery process efficiency, and human capital value) have been evaluated in detail. In a retrospective the value stream approach was perceived positively by the practitioners with respect to process and outcome.

Conclusion: Overall, the conclusion is that this case study indicates that Value Stream Mapping is useful in a software engineering context.

Keywords

1. INTRODUCTION

As global competitiveness and shorter time-to-market come to the software development industry, the search is on for a better way to create first-class software rapidly, repeatedly, and reliably. Lean initiatives in manufacturing, logistics, services, and product development have led to dramatic improvements in cost, quality and delivery time (Dalgobind and Anjani, 2009; Morgan and Liker, 2006; Tong and Li, 2007; Waters and Bevan, 2005); however, the question is do lean practices have the potential to do the same for software development? The short answer is absolutely (see, for example, (Mehta et al., 2008; Petersen and Wohlin, 2010; Poppendieck and Poppendieck, 2008)). Of the many methods that have arisen to improve software engineering processes, Lean is emerging as one that is grounded in decades of work understanding how to make processes better. Lean Software Development provides the theory behind agile software development practices and gives organizations a set of principles from which to fashion software engineering processes that will work best in the context of their customers, their domain, their development capability, and their unique situation (Poppendieck and Poppendieck, 2008). Lean software development has three goals, representing three very different areas of process improvement. They are:

- Creating the right products - Creating product architectures, families, and designs that increase value for all enterprise stakeholders.
- Effective life cycle and enterprise integration – Using lean engineering to create value throughout the product lifecycle and the enterprise.
• Efficient engineering processes - Applying lean thinking to eliminate wastes and improve cycle time and quality in engineering.

“Value Stream Mapping” is one of the seven lean software development tools (McManus, 2005; Mujtaba et al.). “Value stream mapping” is used to achieve the third goal of Lean software development, i.e. efficient engineering processes. A value stream contains all the tasks and actions (both value added and non-value added) currently required to bring a product through the main process steps to the customer a.k.a end-to-end flow of process. The lean philosophy emphasizes to optimize the whole value stream; from the time it receives an order to address a customer’s need until software is deployed and the need is addressed. The reason being if a company focuses on optimizing something less than the entire value stream, the overall value stream is guaranteed to suffer (Poppendieck and Poppendieck, 2008).

The MIT Product Development Value Stream Mapping (PDVSM) Manual provides guidelines as how to do ”value stream mapping” in product development based on the authors’ experiences in Lean engineering within the aerospace industry (McManus, 2005). However, concrete strategies as how to apply, as well as the appropriateness of the applied, the guidelines are missing for the software engineering context, e.g. how to assess the potential value gain from different perspectives of implementing improvement strategies?

Thus, the paper offers two contributions. First, a detailed specification of “value stream mapping” within software development context is presented. “Value stream mapping” steps as prescribed in the MIT manual (McManus, 2005) and in the book by Poppendieck and Poppendieck (Poppendieck and Poppendieck, 2008) are extended to fit the software engineering context. Value aspects to be evaluated, other than lead-time and cost reductions, relevant from different perspectives for software development are presented. Literature lacks guidelines or frameworks to apply “value stream mapping” in software development context. The second contribution is a detailed case study which illustrates how the extended “value stream mapping” framework can be used to select steps and strategies to perform “value stream mapping” and demonstrates usability and usefulness of the extended “value stream mapping”.

The remainder of the paper is structured as follows: Section 2 presents related work focusing on lean in general, and “value stream mapping” application in particular. Section 3 presents the Software Value Map used to evaluate the impact of improvements on different value considerations. Section 4 presents the “value stream mapping” framework for software development. Section 5 explains the case study design used to evaluate the “value stream mapping” framework work. Section 6 illustrates the instantiation of the framework in the case company, as well as the documented outcome of applying the framework. The implications and lessons learned are discussed in Section 7. Section 8 concludes the paper.

2. BACKGROUND AND RELATED WORK

Toyota revolutionized the automobile industry with their approach of “Lean Manufacturing” in the 1980s. Womack and Jones, through publications enhanced the original concept of lean manufacturing into the framework of “Lean Thinking” that gained popularity during the 1990s (cf. (Womack and D. T. Jones, 2007)). Womack and Jones distilled the essence of lean approach into five key principles demonstrating how the lean approach and relevant practices can be extended beyond the automotive industry to any company and organization in any sector. In the early 2000s, Poppendieck and Poppendieck further transferred these principles and practices from the manufacturing domain to specifically suit the software development context and called it “Lean Software Development” (Poppendieck and Poppendieck, 2008).
Rother and Shook (Rother and Shook, 2003) provide a detailed account on the practical implementation of “value stream mapping”, which is also used in other sectors and services related industries. The “Lean Advancements Initiative” research consortium at MIT, has for many years applied lean concepts for improving aerospace product development processes. Their main work “Product Development Value Stream Mapping Manual” (McManus, 2005) offers extensive academic research for learning. Pavnaskar and Gershenson have discussed some differences between manufacturing and engineering and proposed a set of icons, definition of process data and added representation of parallel and sequencing processing (Pavnaskar and Gershenson, 2004). However, explicit strategies as to perform each step of “value stream mapping” are missing both in MIT manual and Pavnaskar and Gershenson work.

Mujtaba et al. applied “value stream mapping” to reduce lead-time in software product customizations and provided some hints and tips as how different steps in “value stream mapping” (drawing current state map, identifying waste etc) can be performed (Mujtaba et al., 2010). The concept of “waste” is the foundation of the lean philosophy. Traditionally waste is defined as any activity that consumes time, resources, or space but does not add value to the product as perceived by the customer. All lean approaches focus on identifying and eliminating waste (see (Petersen and Wohlin, 2011) for further details on the measurement and visualization of the flow of lean software development). However, most of the literature reporting use of “value stream mapping” talks about the seven wastes purely in manufacturing context. Whereas, in the context of software development there can be activities and steps that do not add value directly and thus may be termed as waste. For example, in manufacturing or engineering, performing detailed analysis to ensure architectural flexibility and maintainability for future sustainability might be termed as a waste. Whereas, within the software development context it could be highly value-adding for future additions and modifications consequently reducing lead-time for future releases. Moreover, it is fundamental to realize that a reduction in lead-time might be negatively impacting other value considerations, such as quality of the product and architecture, human capital value, innovation capability etc. Thus, in an effort to eliminate waste extreme caution has to be taken. To the best of our knowledge, this has not been discussed explicitly in existing literature related to “value stream mapping”. Thus, the proposal is to use Software Value Map while analyzing the impact (positive or negative) of eliminating identified wastes. The concept of Software Value Map is introduced in Section 3 and the details as how this can be used within “value stream mapping” are given in Section 4.1.6.

3. Software Value Map

The Software Value Map (Khurum et al., 2011) provides a consolidated view of the software value concept utilizing four major perspectives: the financial, the customer, the internal business process, and the innovation and learning. The value aspects and value components contained in the map are collected through extensive review of economics, management and value-based software engineering literature. The value map offers a unified view of value, which can be used by professionals to develop a common understanding of value, as well as acting as decision support to assure no value perspective is unintentionally overlooked when taking process improvement decisions. The map contains links between different value aspects thus making interrelationships explicit.

The taxonomy used to categorize the perspectives for measuring value was inspired by balanced scorecard (BSC) (Kaplan and Norton 2003; Kaplan and Norton, 2005). BSC can be defined as a set of measures that gives managers a fast but comprehensive view of the business
using four main perspectives, namely the financial, customer, internal process, and innovation and learning, each described below (Kaplan and Norton 2003; Kaplan and Norton, 2005).

The financial perspective contains aspects that address the company’s implementation and execution of its strategy which are contributing to the bottom-line improvement of the company. It represents the long-term strategic objectives of the organization and thus it incorporates the tangible outcomes of the strategy in traditional financial terms (Dolins, 2006; Kaplan and Norton 2003). Some of the most common financial measures that are incorporated in the financial perspective are earned value analysis, revenue growth, costs, profit margins, cash flow, net operating income, and customer value analysis.

The customer perspective defines the value proposition that the company will apply to satisfy customers and thus generate more sales to the most desired (i.e. the most profitable) customer groups (Dolins, 2006). Measures that are selected for the customer perspective should measure both the value that is delivered to the customer (value proposition) with respect to the perceived value, which may involve time, quality, performance and service, and cost, and the outcomes that come as a result of this value proposition (e.g., customer satisfaction and market share).

The internal process perspective is concerned with the processes that create and deliver the customer value proposition. It focuses on all the activities and key processes required in order for the company to excel at providing the value expected by the customers both productively and efficiently. These can include both short-term and long-term objectives as well as incorporating innovative process development in order to stimulate improvement. Quality, cycle time, productivity and cost are some aspects where performance value can be measured (Kaplan and Norton 2003).

The innovation and learning perspective is the basis of any strategy and focuses on the intangible assets of an organization, mainly on the internal skills and capabilities that are required to support the value-creating internal processes. The innovation and learning perspective is concerned with the intellectual capital categorized as human capital, structural capital, and the organization capital of a company (Kaplan and Norton 2003; Liebowitz and Suen, 2000).

BSC combines the financial perspectives with operational perspectives to provide a consolidated view of the business performance and was thus considered as a good base for the categorization in the creation of the Software Value Map (Khurum et al., 2011).

Figure 1 shows an excerpt from the Software Value Map. The categorization of value aspects and value components from various perspectives can be seen in the figure. Looking at Figure 1; “Customer perspective” is one of the four value perspectives. It contains two value aspects (VAs): “VA1.1: Perceived value” and “VA1.2: Customer lifetime value”. Each value aspect further contains one or more sub-value aspects (SVA) or value components (VC). For example, VA1.1 contains three SVAs and two VCs namely: “SVA1.1.1: Intrinsic value”, “SVA1.1.2: Delivery process value”, “VC1.1.3: Network externalities”, “VC1.1.4: Complimentary value” and “SVA1.1.5: User experience value”. Looking further down the structure into “SVA1.1.1: Intrinsic value”, it contains five measurable VCs: “VC1.1.1.1: Functionality”, “VC1.1.1.2: Reliability”, “VC1.1.1.3: Usability”, “VC1.1.1.4: Maintainability” and “VC1.1.1.5: Portability”. The entire Software Value Map together with detailed and complete attributes can be obtained from authors.
4. Value Stream Mapping Framework

In this section, the concept of “value stream mapping” is briefly described. The steps involved in “value stream mapping” are discussed in detail and extensions proposed to existing “value stream mapping” practice are also presented. Each step has several sub-steps that need to be done. “What” needs to be done in each sub-step is described along with the proposals as “how” it can be done.

“Value stream mapping” is based upon lean principles and is a powerful tool that enables identification of opportunities for significant process improvement. “Value stream mapping” assists to uncover bottlenecks in a process that hinders it from flowing at its optimum. It enables organization to understand any workflow with an end-to-end perspective (McManus, 2005; Srambikal, 2008; Vujica Herzog et al., 2008).

Figure 2 shows the notations used for drawing the “value stream mapping”. These notations have been used by MIT manual (McManus, 2005) to demonstrate how “value stream mapping” is applied to product development. A “Process Box” indicates a single step or activity in the process in which the work is flowing. The step is written inside the box for example, “Specify Requirement”. An “Arrow” indicates the direction of flow from one step to another. The “Timeline” indicates the sequence of process steps arranged in a chronological order with different times displayed along a line (drawn from left to right). The different times shown in Figure 2 represent different measurements, which are described below. The “burst signals” represent alarm signals indicating waste. For example, long waiting times with high improvement potential can be highlighted. Different measurements can be used in “value stream mapping” (as detailed in Section 4.2.3) but the ones shown in Figure 2 are related to time. “Processing time
(PT)”: the time taken by one or several persons within the process step or activity to do the work.
“Waiting time (WT)”: the average wait time within a process step or activity or between sub-processes.

Figure 2 – Value Stream Map Notation

Steps and sub-steps of the Value Stream Mapping process are shown in Figure 3. Here it can be seen that the additions proposed by this paper are clearly marked with a star (“*”), indicating the extension to the “value stream mapping”. In the following text, proposed extensions and strategies are also marked with (“*”) to highlight the contribution.

Figure 3 – Value Stream Mapping Process
4.1 S1: Getting Started

The first step includes all activities that are related to preparation and planning of the “value stream mapping” activity.

4.1.1 S1.1: Identifying Key Stakeholders

What: Stakeholders can be defined as the “people who derive value of any sort from the process” (McManus, 2005). If the stakeholders are not identified, it will not be possible to elicit and reconcile purpose of the “value stream mapping” (S1.2), put together right team (S1.3) and define value in a useful way (S1.5). The key stakeholders, and their expectations for the process, its outputs, and the improvement of both, all need to be identified.

How: The MIT manual (McManus, 2005) suggests some types of stakeholders that might be relevant for the current “value stream mapping”. These are given in the Table 1.

Table 1 - Types of Stakeholders for Value Stream Mapping (McManus, 2005)

<table>
<thead>
<tr>
<th>Stakeholder types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct internal customers</td>
<td>The users of the output of the process (that is to be mapped), the customer of the design or other final product development outputs</td>
</tr>
<tr>
<td>Indirect internal customers</td>
<td>Participants in the processes further down the value stream</td>
</tr>
<tr>
<td>External customers</td>
<td>The users of the physical product that will be made from the design</td>
</tr>
<tr>
<td>Others</td>
<td>Suppliers of information or resources to the product development process</td>
</tr>
<tr>
<td></td>
<td>_ Suppliers of parts or assemblies to the physical product</td>
</tr>
<tr>
<td></td>
<td>_ The firm or enterprise management or leadership</td>
</tr>
<tr>
<td></td>
<td>_ Employees or their representatives</td>
</tr>
<tr>
<td></td>
<td>_ The community in which the process takes place</td>
</tr>
<tr>
<td></td>
<td>_ The general tax paying public</td>
</tr>
</tbody>
</table>

4.1.2 S1.2: Defining the purpose

Once the key stakeholders have been identified it is central to elicit the main purpose of applying “value stream mapping”.

1. Understand and communicate a process: When different parts of a process are mapped through the value stream, this process in itself is beneficial to for the understanding of the tasks and information flows. “Value stream mapping” provides a great form of visual communication among everyone involved.

2. To evaluate a concrete improvement: Some improvement to an existing process has been already suggested and the aim is to evaluate it.

3. Optimize towards a specific target: Mostly companies have some themes or targets to be achieved biannually or quarterly, for example, to reduce lead-times or improve performance quality of the product by 10% (Michael, 2011). Thus, “value stream mapping” can be used to optimize existing processes to achieve the set targets.
4.1.3 S1.3: Defining the team

What: If the problem in hand is simple, usually one person team is enough. However, in real software development scenarios this is seldom the case. Thus, a variety of perspectives are needed to reveal the otherwise difficult-to-see software development value stream for achieving the targeted purpose (as identified in S1.2). Therefore, the team should embody a balance of different relevant perspectives.

How: The MIT manual (McManus, 2005) suggests coverage of different perspectives (role and authority wise).

- Product perspective: is needed to include roles that supply or use the output of the process. There might be wastes in the produced product (for example, extra functionality, unnecessary documentation) which can be revealed more accurately by the roles using the products of the process.

- Process perspective: is needed to include roles involved in the process. This perspective is needed to identify wastes related to the process (for example, time delays). Within process perspective, it is important to have chefs and process doers within each role because: (1) Process chefs have the knowledge and experience in the process to be improved, and have the authority to change it; (2) Process doers are the ones with detailed, on-the-ground knowledge of the process, its strengths and weaknesses, and how it is currently executed. They are the best source of both improvement ideas and knowledge of the possible pitfalls of process change. If they are not included, one might only be scratching the surface of the problems/issues actually faced by the process doers.

With respect to a typical software development process, a classification of perspectives and relevant roles are presented in Table 2*. It is recommended that one person per cell should be ideally included in the team.

<table>
<thead>
<tr>
<th>Role</th>
<th>Perspective</th>
<th>Product</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Chefs</td>
<td>Doers</td>
</tr>
<tr>
<td>Strategic management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture and design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivery and release</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The team participants can be selected depending on the targeted purpose. For example, if the purpose is to evaluate a concrete improvement then it might be enough to have the customers, suppliers of the product and process doers as participants. The reason being that the improvement has been already suggested and approved by the management thus; the process chefs need not be included. However, if the purpose is to optimize towards a target then there is a need to include
process chefs to approve actions for eliminating wastes. However, there are certain limitations while defining a team:

- A team should not comprise more than 12 people because that can lead to extended and unnecessary discussions if workshops are held to perform S1.3 and S1.4 of the “value stream mapping”.
- Formal commitment from upper management and roles involved are a pre-requisite for the success of “value stream mapping” exercise.

4.1.4 S1.4: Training the team

**What:** The selected team must be at least minimally trained in lean and the methods and tools chosen for “value stream mapping” and analysis.

**How:** A half-day session on lean thinking, “value stream mapping” concepts and some translation to product development problems should be conducted (McManus, 2005).

4.1.5 S1.5: Bounding the problem

**What:** Following critical elements of the value stream need to be explicitly defined before embarking on the journey of “value stream mapping” (Poppendieck and Poppendieck, 2008).

1. The *process bounds* including the beginning and ending points of the process, and its organizational boundaries.
2. The product or products on which the process operates.
3. The owner who will provide the point for direct responsibility for the stream, whether this be a group or an individual.
4. The output that provides reason for the stream to exist - it is the packaged value generated by the given process. The customers then receive the product from the owner at the end of the value stream. These customers do not necessarily represent someone external to the organization; they may include internal customers.
5. The initial inputs as well as the additional knowledge and information that may be pulled into the process – on which the process operates.
6. The constraints that place limits on the process.

**How**: The MIT manual (McManus, 2005) does not provide any guidelines as how to elicit the above mentioned critical elements for binding the problem. Our contribution is the suggested strategies below that can be employed to bind the problem. These strategies are discussed with respect to the tradeoffs between accuracy, time and resources required and complexity of the problem in hand.

- **Documentation analysis:** The organizer of the “value stream mapping” initiative can bind the problem and specify the above mentioned elements by studying the available documentation. This can save time and effort (as shown in Table 3) however, documented processes may not contain implicit products, customers, constraints etc.
- **Interview with experienced team members:** Two or three experienced team members having an overview of the entire process can be interviewed to bind the problem and
identify the above elements. However, this is more time and resources intensive than documentation analysis but can be more accurate (as discussed in Table 3).

- **Documentation analysis and discussion:** A reasonable option can be that the organizer binds the problem through documentation analysis and subsequently discusses with one or two team members who have extensive knowledge and experience of the entire process.

- **Workshop with team members:** If the problem in itself is very complex and accurate binding is required, a half-day workshop can be conducted with all the team members to explicitly identify the above mentioned critical elements.

An overview of accuracy, time and resources, and capability to handle complexity is summarized for bounding the problem is given in Table 3. Definitions of the scales: very high, high, medium, and low are presented in Appendix A.

**Table 3** - Comparison of strategies for binding the problem

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Time and resources</th>
<th>Capability to handle complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation analysis</td>
<td><strong>Low to medium to high</strong> depending on the documentation quality and up-to-dateness, high risk that the underlying problem is not revealed (e.g. it is not clear where a process begins/ends)</td>
<td><strong>Low</strong> as only the organizer is involved, not binding organizational resources</td>
<td><strong>Low to medium to high</strong> depending on the documentation quality and up-to-dateness (e.g. if the end to end process and relationships are captured well enough)</td>
</tr>
<tr>
<td>Interviews</td>
<td><strong>Medium to high</strong> as a sub-set of team members is being interviewed individually giving very detailed information and the opportunity to as follow-up questions, but a complete and integrated picture of the problem is not available</td>
<td><strong>Medium</strong> as only few members of the selected team are consulted</td>
<td><strong>Medium to high</strong> as the interview allows to ask follow-up questions and getting detailed information, but the information is not easy to integrate in a holistic view of the problem scope</td>
</tr>
<tr>
<td>Documentation analysis and discussion</td>
<td><strong>Medium to high</strong> as the document analysis is complemented by discussions which increases accuracy (triangulation)</td>
<td><strong>High</strong> because the effort combines the efforts from documentation analysis and interviews.</td>
<td><strong>Medium to high</strong> as the analysis depends on the documentation quality and up-to-dateness, which can be improved through complementary interviews.</td>
</tr>
<tr>
<td>Workshop with team members</td>
<td><strong>Very high</strong> as complete picture of the problem will be available because all the key team members are participating</td>
<td><strong>High</strong> as all team members have to be present at one occasion.</td>
<td><strong>Very high</strong> as team members’ views can be combined to get a picture of the end to end view of the problem scope</td>
</tr>
</tbody>
</table>
4.1.6 S1.6: Defining and understanding value creation*

**What**: Defining the value of software product development work in a general way is an ongoing, and probably never-ending, task. It is fundamental to understand that without a working definition of the value created by the process being mapped, and an appreciation for how that value is created, one cannot guide an improvement effort. Here value has to be understood in two different contexts; (1) the value of the process output to the larger enterprise and (2) the creation of value during the carrying out of the individual tasks that make up the process.

**How**: If the context is the value of the process output to the larger enterprise the goal generally is to carry out the process well, efficiently and in a timely fashion. This goal can be made explicit using the following formulation:

Produce the *required outputs*, *without defects*, as *efficiently* as possible, and at the *right time*.

For example, a goal for a software development company can be:

Develop *features usable* by customers, *without defects*, as *efficiently* as possible, at the *right time*.

The MIT manual (McManus, 2005) suggests some metrics to measure time (for example, cycle time and lead-time) and costs (for example, recurring and non-recurring costs). Time and cost are important value considerations. However, these are only project focused short-term value considerations and product or organization focused long-term value considerations are largely missing in the MIT manual. For example, customer satisfaction is one of them. The MIT manual does talk about customer satisfaction, however it does not break it down to what actually customer satisfaction means and how it can be evaluated. Chase proposed a list of aspects of value that a task could contribute (Chase, 2001). However, a detailed account of value considerations relevant for different perspectives like customer and internal business value are missing. Thus, to fill this gap the proposal is to use the Software Value Map” (see Section 3) by Khurum et. al. (Khurum et al., 2011) to identify value considerations relevant for the goal to be achieved.

In order to identify value components relevant to the stated goal, a **Goal Question Value** approach is proposed. It is similar to the goal question metrics approach but metrics are replaced with value in the given context because quantitative measurement of the value of individual tasks has proven elusive (McManus, 2005). Moreover, intangible value components like customer satisfaction are difficult to be measured directly in numbers (McManus, 2005) either financially or in any other metric. Therefore, instead of identifying metrics to be measured against questions for progress evaluation towards the goal, value components relevant for the goal to be achieved are evaluated. The process is simple and straightforward.

1. Pick an italicized phrase/word from the goal.
2. Against the selected italicized/word, take each value perspective given in the Software Value Map and go through all the value components given in that value perspective to identify value components that need to be evaluated. For example, features usable by customers from customer’s value perspective means the product’s intrinsic functionality and usability value components (see Figure 1) need to be evaluated. In this case other perspectives like internal business value, innovation value are not relevant so they can be skipped.
3. Repeat instructions given in bullet 1 and bullet 2 until value components relevant for all the italicized phrases/words from the goal have been identified. A detailed account of this process is given in the case study (Section 6). Not going into the details of quantification, a five-point rubrics based evaluation criteria has been designed for evaluating a value consideration. For example, to evaluate usability value for the customer, the rubric choices are shown in Table 4.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>Greatly improves usability of the software product</td>
</tr>
<tr>
<td>+1</td>
<td>Improves usability of the software product</td>
</tr>
<tr>
<td>0</td>
<td>No affect</td>
</tr>
<tr>
<td>-1</td>
<td>Degrades usability of the software product</td>
</tr>
<tr>
<td>-2</td>
<td>Seriously degrades usability of the software product</td>
</tr>
</tbody>
</table>

In the Software Value Map (Khurum et al., 2011), usability value has been further decomposed into understandability value, learnability value and operability value. Depending on the level of evaluation required, one can use value considerations at an abstract or at a more detailed level.

4.2 S2: Mapping the current value stream map

In S2 of the “value stream mapping” the current state map is created. The map is the basis for identifying wastes and reasons for the occurrence of waste. The most critical wastes will then be addressed through process improvement actions in S4.

4.2.1 S2.1: The Zeroth map

**What:** A Zeroth map, i.e. representation of the process with a high-level mapping, can help to provide a good initial “big picture” of the process. These high level maps should be used to help visualize the process and prepare for a more detailed analysis. It is unlikely, however, that at this level wastes will become apparent (McManus, 2005).

**How:** Existing gantt charts, ward maps or work breakdown structures can be used to get an idea. For more details, see MIT manual (McManus, 2005).

4.2.2 S2.2: Tasks and flow

**What:** Here the undertaking is to identify tasks that make up the process, and the basic flows of information between them. This map will establish the topology of the value stream: linear, iterative or spaghetti-like? The type of map to be used, and its symbols and conventions, will depend on the answer to these questions.

**How:** Our contribution is the suggested strategies (from 1-4) below that can be employed to bind the problem. MIT manual also suggests a strategy (see bullet point 5) which could give a detailed and accurate account of the tasks and flows however, the cost and time required to follow this strategy are very high. The strategies are discussed with respect to the tradeoffs between accuracy, time and resources availability and complexity of the problem in hand.
1. Documentation analysis: Zeroth map drawn or obtained can be used to identify tasks and flows. If the map contains all the necessary information, it would be easier to identify tasks and flows.

2. Interview with experienced team members: Two or three experienced team members having an overview of the entire process can be interviewed to identify tasks and flows between them. However, this is more time and resources intensive than documentation analysis but can be more accurate.

3. Documentation analysis and discussion: A reasonable option can be that the organizer identifies tasks and flows using Zeroth map and subsequently discusses and updates the maps by having interviews with one or two team members who have extensive knowledge and experience.

4. Workshop with team members: If the process in itself is very complex and has many interdependencies and iterations, a half-day workshop can be conducted with all the team members to create the Zeroth map.

5. Follow the work: The best option, as suggested in the MIT manual (McManus, 2005), is to uncover the true value stream bystrapping oneself to the product. If this is not possible, one must strap himself/herself to the information i.e. the work package that starts with the process inputs and accumulates and transforms until it becomes the process output. One can start from the input to the process and work forward by continuously asking “what happened next” or one can start from the conclusion of the process and trace back to the start. In complex flows with interdependencies and iterations both might be required. If physical walking is not possible (because some tasks are done remotely), one can talk to the participants by the highest-bandwidth channel available (e.g., telephone in preference to email) and make them send back real examples of the documents or other information products they produce and handle. However, being actually present during the tasks can increase the threat of altering the subject of one’s study. Moreover, one needs to be cautious of an additional threat that a single case being studied might not be representing a typical case.

An overview of alternative methods with respect to accuracy, time and resources, and the capability to handle complexity is provided in Table 5. Based on available resources, time constraints, accuracy required and complexity of the process, one of the above mentioned strategies can be chosen.
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Time and resources</th>
<th>Capability to handle complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation analysis</td>
<td><strong>Low to medium</strong> to high depending on the documentation quality (e.g. tasks missing) and up-to-dateness (e.g. documentation erosion)</td>
<td>Low because only the organizer studies process documentation and maps the process steps without involvement of team members.</td>
<td><strong>Low</strong> to medium as highly dependent on the documentation quality and up-to-dateness</td>
</tr>
<tr>
<td>Interviews</td>
<td><strong>Medium to high</strong> as a sub-set of team members is being interviewed individually and complete picture of the process steps/interfaces between processes is not available</td>
<td><strong>Medium</strong> because only a sub-set of team members is interviewed</td>
<td><strong>Medium</strong> to high as one is getting an in-depth understanding of the tasks of individuals, but no immediate feedback on how these are connected and embedded in the end to end process flow</td>
</tr>
<tr>
<td>Documentation analysis and discussion</td>
<td><strong>Medium to high</strong> as the document analysis (e.g. study of process maps) is completed by discussions (e.g. check whether process map matches current reality with practitioner) which increases accuracy (triangulation)</td>
<td><strong>High</strong> because the effort combines the efforts from documentation analysis and interviews</td>
<td><strong>Medium to high</strong> because the analysis depends on the documentation quality (e.g. sub-processes are described, but not their interactions and artifacts). The information can be added, however, individual interviews do often not reveal a holistic picture end to end.</td>
</tr>
<tr>
<td>Workshop with team members</td>
<td><strong>Very High</strong> as complete picture of the process and tasks and flows will be available since all the key stakeholders are participating and can complement their views on how the process is actually working</td>
<td><strong>High</strong> because all team members have to be present at one occasion</td>
<td><strong>Very High</strong> as complex interactions between process steps and a holistic view becomes visible through interaction and feedback.</td>
</tr>
<tr>
<td>Follow the work</td>
<td><strong>Very high</strong> as the actual work process and produced artifacts (e.g. requirements, design, etc.) can be observed in the real setting</td>
<td><strong>Very high</strong> because the organizer of the value stream has to involve many people throughout the process and disturb them during process execution, the observation might have to be done over a long period of time</td>
<td><strong>Very high</strong> because the end-to-end process with all interactions/artifacts are captured by following the work product, especially regarding the interfaces between different sub-process activities</td>
</tr>
</tbody>
</table>
4.2.3 S2.3: Data collection

**What:** The purpose of Step 2.3 is to collect data for each task that can help to put alarms and identify wastes. Collecting and using data requires a balance between the effort involved and the payoff expected. Although collecting task data is difficult, it is also possible to collect too much of it. The point is to achieve a lean process, not to build a complete model of the process. Depending on the purpose of “value stream mapping”, different data needs to be collected. In Table 6 different information needs and measurements are presented depending on the purpose of the value stream map (McManus, 2005).

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Data related to Example</th>
<th>Metrics</th>
</tr>
</thead>
</table>
| Communication and understanding | - Task  
- Waiting  
- Information | Cycle time, rework costs |
| Evaluating concrete improvement | Concrete improvement. It could be:  
- Task  
- Waiting  
- Information related  
- Artifact (information, software product) | If improvement suggested was to improve external quality of the product then product quality attributes can be evaluated (given in the Software Value Map). If suggestion is to improve requirements specification usability, then understandability, ease of use of the new format, completeness of specification etc will be measured |
| Optimizing value considerations derived from the goal | It could be:  
- Task  
- Waiting  
- Information related  
- Artifact (information, software product itself) | For example, in order to reduce lead-time, waiting time data will be collected to identify if there is a waste. However, it is important to carefully inspect if data related to other value considerations need to be collected as well to ensure that root cause of longer lead-time is not something else. For example, if a bad quality product is produced, testing and removing defects can take a longer time and in this case, wait time is not an issue rather inefficient process related to quality assurance are the culprits. This is also in line with suggestion by Trischler to avoid the tendency to start with a preexisting theory and collect the data necessary to validate it (Trischler, 1996). |

**How:** Different strategies can be adopted to collect required data.

1. **Quantitative analysis of historical data:** Mujtaba et. al (Mujtaba et al., 2010) suggested if historical data is available within a company it can be readily used. For example, if data about lead-time (LT): average time it takes for one request to go through the entire process from start to finish including all queuing/waiting times between is to be collected; each request can be individually examined and time stamps for the start/end of each activity in the current “value stream mapping” can be recorded. If some data is missing, one can look for the end date of the preceding activity or start date of the next immediate activity to determine the missing start/end date.

2. **Interview with experienced team members:** In case the required data is not available, Poppendieck and Poppendieck assert that the lead measurements for “value stream mapping”
can be based on guesstimates or expert opinion (Poppendieck and Poppendieck, 2008). Two or three experienced team members having an overview of the entire process can be interviewed to provide approximate measurements. However, this is more time and resources intensive than documentation analysis but can be more accurate. Another issue with this strategy as pointed out by Mujtaba et al. (Mujtaba et al., 2010) is that there could be a lot of variation among interviewees’ responses which can make it difficult to get a realistic average or approximation. The next strategy can be seen as a solution to this problem.

3. **Documentation analysis and discussion**: A reasonable tradeoff between the two strategies can be that organizer can collect required data if available historically and subsequently discusses and updates with one or two team members who have extensive knowledge and experience. This can provide data triangulation using multiple sources.

4. **Workshop with team members**: If historical data is not available and/or a lot of variation can exist among interviewees’ responses, a half day workshop can be conducted with all the team members to explicitly collect data. And since it will be a joint workshop, the variations can be picked and discussed to find a realistic average or approximation. Collection of data in a workshop can be combined with the workshop where tasks and flows are identified (if workshop is organized to identify tasks and flows). An overview of alternative methods with respect to accuracy, time and resources, and the capability to handle complexity is provided in Table 7.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Time and resources</th>
<th>Capability to handle complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation analysis</td>
<td>Low to medium</td>
<td>Low to medium</td>
<td>Could be low to medium to high depending on the documentation quality and up-to-dateness. For example, lead-time data might be outdated and not reflecting the current process.</td>
</tr>
<tr>
<td></td>
<td>as a sub-set of team members is being interviewed individually and variations between interviewees can exist (as reported by Mujtaba et al. (Mujtaba et al., 2010))</td>
<td>Medium</td>
<td>as only individual team members are interviewed for their opinions, which is a risk as no agreement on the current situation is given.</td>
</tr>
<tr>
<td></td>
<td>Medium to high</td>
<td>High</td>
<td>Medium as times in-between activities are often not known by individuals (e.g. the receiver of an artifact knows that he/she has waiting time, but the sender is not aware of that)</td>
</tr>
<tr>
<td></td>
<td>as the document analysis (e.g. identification of lead-times) is complemented by discussions (e.g. interviews)</td>
<td>High</td>
<td>Medium to high because the analysis depends on the quality of the collected data. Also, the collected data informs interviewees in</td>
</tr>
<tr>
<td>Method</td>
<td>Accuracy</td>
<td>Time and resources</td>
<td>Capability to handle complexity</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Workshop with team members</td>
<td><strong>Very high</strong> as multiple opinions about measurements (e.g. lead-time) become obvious and can be resolved and agreed on in the team easily</td>
<td><strong>High</strong> as all team members have to be present at one occasion</td>
<td><strong>Very High</strong> as the end-to-end view of measurements gives a good combined picture of the overall process end-to-end.</td>
</tr>
<tr>
<td>Follow the work</td>
<td><strong>Medium to very high</strong> as actual data is collected by following one item through the flow. However, the item followed might not be a good representative for a specific measurement, e.g. lead-time, which leads to variance in accuracy.</td>
<td><strong>Very high</strong> because the organizer of the value stream analysis has to involve many people throughout the process and disturb them during process execution, the observation might have to be done over a long period of time</td>
<td><strong>Very high</strong> because the end to end process with all interactions/artifacts is captured by following the work product from beginning to end.</td>
</tr>
</tbody>
</table>

### 4.2.4 S2.4: Evaluation of value

**What:** In this step, value-added at each task is assessed.

**How:** Traditional recommendation of categorizing each task as value-added, necessarily non-value-added, pure waste is difficult just by looking at the map as most non-value adding activities might be hiding within value adding tasks (McManus, 2005). Thus, MIT manual provides following recommendations to evaluate value:

- Evaluate how each task adds value. Does it add to the product definition? The process definition? Does it reduce uncertainty in ways that make the definitions more valuable?
- Value assessments can also be made of the information flows between the processes. These assessments are best done by those on the receiving end of the data flow.
- Again, the first check is for non-value-added information; if useless information is being produced and propagated this is pure waste. More likely the information is necessary, but it may not be of good quality due to incompleteness, errors, or formatting problems. A simple relative ranking with respect to perceived completeness, quality and format of data may be quite useful in many cases.
- Value information can be displayed on the map in a variety of ways. Tasks can be tagged as Value Added (VA), Required Non-Value-Added (R-NVA) or Non-Value-Added (NVA). If only a few NVA tasks are evident, they should be highlighted by burst symbols. Types of value contributions can be identified by acronym codes, color, hatching or other symbols.

However, the researchers argue the MIT recommendations are inadequate; due to the fact that different perspectives for value evaluations are ignored. For example, a certain task can be
adding pure business value (for example, improving product architecture maintainability) which externally does not have visible value from the customer perspective. According to the MIT recommendations it should be regarded as non-value adding as it does not add to a product or process definition or reduce uncertainty in any apparent way. Whereas, from the business’s sustainability perspective, it improves product’s architecture sustainability by making it maintainable. Therefore, this task should not be regarded as non-value-adding or pure waste. In traditional manufacturing, a task adds value directly to a process or product whereas in a software development context, there can be several tasks that do not add direct the product or process value (as exemplified above) however; they do add value to the business by making the product more sustainable. Thus, our recommendation from a software product development perspective is to evaluate value from all the four perspectives: customer, internal business, innovation and financial (value aspects and value components that can be evaluated within each perspective are given in the software value map (Khurum et al., 2011)).

Moreover, MIT manual does not guide as how to systematically deduce values that need to be evaluated as they can grow easily due to a general tendency to do evaluations from different perspectives and angles. GQV approach is proposed to deduce value considerations that need to be evaluated. This would keep value considerations to be evaluated focused (based on goals) and manageable.

4.2.5 S2.5: Understanding Iteration

What: It is quite common in case of product development that “value stream mapping” will have branching and iterative information flows (McManus, 2005). This might result in a lot of complexity and the effectiveness of the tool to visualize the process flow will diminish. For such scenarios, a Design Structure Matrix (DSM) can be used (McManus, 2005).

How: The DSM is a square matrix with N rows and columns, one for each of N tasks in the process. Information flow between the tasks is depicted by dots in the cells of the matrix. If task $m$ provides information to task $n$, a dot is placed in the cell intersecting column $m$, row $n$. One hopes that information would flow from earlier tasks to later ones. If the tasks are arranged in nominally chronological order, having dots only below the diagonal means information is flowing only from early tasks to later ones. A dot above the diagonal represents information flowing from a later task to an earlier one in iteration. The further away from the diagonal the dot is, the longer the iteration loop processes running independently in parallel, will be. Other facts of the information flow, like branching and recombining, processes that are fully coupled to each other, or processes running independently in parallel, can all be seen at a quick glance with this matrix. However by displaying information with this of matrix, a lot of richness of the process map might be comprised. But on the other hand, the complex information flows can be see and analyzed in a compact and concise way (McManus, 2005).

4.3 S3: Identifying the waste

In this step the waste (e.g. long waiting times) and the reasons/root causes for the waste are identified and documented.

What: Once there is a clear vision of the process and value added by the tasks within the process, the next logical step is to proceed towards its improvement (step 4, see Figure 3). Prior to that, a rudimentary step is elimination of waste (S3) from the process in all of its forms. The sources of waste in product development processes are many and varied, thus it is not possible to propose a cookbook for its identification and elimination. There are seven different types of info
wastes discussed within lean manufacturing context (Shingo and Dillon, 1989). Poppendieck and Poppendieck have translated these wastes to fit into software development context (Poppendieck and Poppendieck, 2008). These wastes along with examples are presented in Table 8.

Table 8 - The Seven Wastes with Examples (Poppendieck and Poppendieck, 2008)

<table>
<thead>
<tr>
<th>Manufacturing</th>
<th>Software Development</th>
<th>Examples</th>
</tr>
</thead>
</table>
| In-process inventory| Partially done work  | • Uncoded documentation  
|                     |                      | • Unsynchronized code  
|                     |                      | • Untested code  
|                     |                      | • Undocumented code  
|                     |                      | • Undeployed code  |
| Over-production     | Extra features       | • Functionality that is not required by the customer  
|                     |                      | • Features for which markets are not ready  |
| Extra processing    | Relearning           | • Making a decision about a requirement (to delay its implementation) and later forget why that decision was made  
|                     |                      | • Making a decision and forgetting the rationale because it was not documented  |
| Transportation      | Handoffs             | • Difficulty to transfer tacit knowledge (for example, design decisions and rationale)  
|                     |                      | • Incompatible information types (drawings vs. digital descriptions)  
|                     |                      | • Incompatible software systems or tools  
|                     |                      | • Lack of availability, knowledge, or training in conversion and linking systems  |
| Motion              | Task switching       | • A developer doing both new software development and maintenance of the previous version  
|                     |                      | • Lack of on-line access  
|                     |                      | • Lack of digital versions of historical information  
|                     |                      | • Team members not co-located  
|                     |                      | • Organization structure inhibits formation of right teams  |
| Waiting             | Delays               | • People waiting for information: Waiting for getting complete requirements from  
|                     |                      | • the customer  
|                     |                      | • Waiting for months for project approval  
|                     |                      | • Waiting for resources to be assigned  
|                     |                      | • Waiting for the whole system to be done before one can get the key features that he/she really needs  
|                     |                      | • Information waiting for people: detailed requirements specification created upfront  |
| Defects             | Defects              | • Leaving testing towards the end  
|                     |                      | • Not finding defects as early as possible  
|                     |                      | • Lack of disciplined reviews, tests, verification  |

**How**: The MIT manual does not prescribe as how to identify wastes. The researchers contribute by collecting different strategies to identify these wastes. After the current state map is completed, one of the following strategies can be used to identify waste.

1. **Quantitative analysis of historical data**: If quantitative historical data is available it can be used to identify wastes. For example, if data about lead-time (LT): average time it takes for
one request to go through the entire process from start to finish including all queuing/waiting times between is to be collected; each request can be individually examined and time stamps for the start/end of each activity in the current "value stream mapping" can be recorded (Mujtaba et al., 2010). If some data is missing, one can look for the end date of proceeding activity or start date of the next immediate activity to determine the missing start/end date. If there is a waiting time between the end of one task and start of another task, this could be potential delay waste.

2. **Interview with experienced team members:** In case the required data is not available, expert opinion can be used to identify wastes. Two or three experienced team members having an overview of the entire process can be interviewed to identify wastes. However, this is more time and resources intensive than documentation analysis but can be more accurate. On the other hand, since only two or three people involved in the process are interviewed the complete perspective might be missing and thus some wastes might be left unidentified or misidentified. The next strategy can be seen a solution to address this issue.

3. **Documentation analysis and discussion:** A reasonable tradeoff between the two strategies can be that organizer can identify wastes based on the available historical data and then discusses and updates with one or two team members who have extensive knowledge and experience. This can provide data triangulation using multiple sources.

4. **Workshop with team members:** If historical data is not available and/or there is a greater risk of missing actual wastes to be identified, then a workshop can be conducted with all the team members to explicitly collect data. And since it will be joint workshop, the variations can be picked and discussed to find actual wastes. If workshop strategy is used to draw current state map, identification of waste can be done as a second step of the same workshop instead of arranging it at a different date and time.

5. **Follow the work:** The best option to uncover the wastes would be to strap one’s self to the product (McManus, 2005). If this is not possible, one must strap himself/herself to the information the work package that starts with the process inputs and accumulates and transforms until it becomes the process output. One can start from the input to the process and work forward by continuously observing “what is happening” or one can start from the conclusion of the process and trace back to the start. In complex flows with interdependencies and iterations both might be required. Try physically walking through the process, talking to the participants (ideally these are the ones in the team as well), and touch the information created. If physical walking is not possible (because some tasks are done remotely), talk to the participants by the highest-bandwidth channel available (e.g., telephone in preference to email) and make them send real examples of the documents or other information products they produce and handle. However, being actually present during the tasks can increase the threat of altering the subject of your study by your presence. One needs to be cautious of an additional threat that a single case being studied might not be representing a typical case. If used carefully, this strategy can be very beneficial to reveal some of the most critical wastes which could not be identified by the people involved as they do not look at the process through external lens.

For the identification of waste the strategies of document analysis, interviews, documentation analysis and discussion, workshop, and follow the work are compared with respect to accuracy, time and resources, and capability to handle complexity, see Table 9.
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Time and resources</th>
<th>Capability to handle complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation analysis</td>
<td><strong>Low</strong> as documentation often does not reveal the priority in waste (e.g. which one is a problem and which one justified), and the reasons for waste are hidden.</td>
<td><strong>Low</strong> as no practitioner/team-member has to be involved</td>
<td><strong>Low</strong> because documentation seldom reveals cause-effect (e.g. reason in long lead-times in design is due to requirements not being well written)</td>
</tr>
<tr>
<td>Interviews</td>
<td><strong>Low to medium</strong> as a sub-set of team members is being interviewed individually and an overall perspective might be overlooked by the interviewees. However, in-depth information can be elicited through follow-up questions.</td>
<td><strong>Medium</strong> as only individual team members are interviewed for their opinions, which is a risk as no agreement on the current situation is given.</td>
<td><strong>Medium to high</strong> as individuals often do not have an overview of the cause effect relations from an end-to-end perspective. However, follow-up questions can be used to elicit detailed descriptions of their wastes and reasons for them.</td>
</tr>
<tr>
<td>Documentation analysis and discussion</td>
<td><strong>Medium to high</strong> as document analysis is complemented by the views of individuals.</td>
<td><strong>High</strong> because the effort combines the efforts from documentation analysis and interviews</td>
<td><strong>Low to medium</strong> because documentation often hides complex relationships as those cannot be easily expressed, and only individual opinions are used to complement the documentation (i.e. holistic/interactive view of the process is hard to capture)</td>
</tr>
<tr>
<td>Workshop with team Members</td>
<td><strong>Very high</strong> as complete picture of the process will be available and different perspectives can be taken into consideration, the perspectives providing feedback to each other</td>
<td><strong>High</strong> as all team members have to be present at one occasion</td>
<td><strong>Very high</strong> because the interactions between different parts of the process can be discussed when representatives of each relevant part are present (e.g. the impact of the quality of requirements on design and test)</td>
</tr>
<tr>
<td>Follow the work</td>
<td><strong>Medium to very high</strong> because waste can be observed by investigating the actual process in execution.</td>
<td><strong>Very high</strong> because the organizer of the value stream analysis has to involve many people throughout the process and</td>
<td><strong>Very high</strong> as the end to end process and its interactions are observed, the observer will be able to recognize impacts of</td>
</tr>
</tbody>
</table>
4.4 S4: Improve the process

In this step improvement alternatives are suggested and the effect of the improvement alternatives are documented in the future state map.

4.4.1 Step 4.1: Strategies to eliminate the waste

What: Once the wastes have been identified, one or several strategies can be applied to eliminate the identified wastes.

How: A list of strategies that can be applied to eliminate/reduce each type of waste is listed in Table 10. A detailed description of these strategies is available in MIT manual (McManus, 2005) and in Poppendieck and Poppendieck (Poppendieck and Poppendieck, 2008). Note that this list is not exhaustive and there could be other strategies to reduce/eliminate wastes. However, this list can be used as a first input to identify strategies for eliminating/reducing identified wastes.

<table>
<thead>
<tr>
<th>Waste Type</th>
<th>Possible Strategies</th>
</tr>
</thead>
</table>
| Partially done work | • Adopt just in time principle when preparing artifacts. Write requirements when they are needed to be specified (no sooner, No later)  
• Workspaces and parallel codes should be synchronized as frequently as possible  
• Either code is integrated, tested and accepted or it does not count  
• Deploy code as frequently as possible. |
| Extra features      | • Focus on what is really needed by the customer to get the job done  
• Be strongly biased against adding a feature  
• Carefully analyze the need for architecture flexibility versus the effort to create architectural flexibility |
| Relearning          | • Have short and concise documentation where needed: use as few words as possible, use figures, graphs and tables instead. Everything must fit on one side of A3 |
| Handoffs            | • Reduce the number of handoffs  
• Use cross-functional teams so that people can teach each other how to ride  
• Use high bandwidth communication (face-to-face discussion, direct observation, interaction with mockups, prototypes and simulations) |
| Task switching      | • Have two people rotate off the team every month or every iteration to handle all maintenance for the duration  
• Shared work experiences: Set aside two hours each morning where the
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>team can jointly handle all issues that developed over the last 24 hours</td>
</tr>
<tr>
<td></td>
<td>- Aggressively triage support request and do only urgent work on daily basis</td>
</tr>
<tr>
<td></td>
<td>- Keep all customers on a single code base and release a new version every week</td>
</tr>
<tr>
<td>Delays</td>
<td>- Establish an average or pseudo takt time</td>
</tr>
<tr>
<td></td>
<td>- Time coordination</td>
</tr>
<tr>
<td></td>
<td>- Assure availability of information: make information visual, make information flow physically, pull and do not push information</td>
</tr>
<tr>
<td></td>
<td>- Eliminate bottlenecks</td>
</tr>
<tr>
<td>Defects</td>
<td>- Assure the timely availability of resources</td>
</tr>
<tr>
<td></td>
<td>- Minimize and buffer variation</td>
</tr>
<tr>
<td></td>
<td>- Clear external causes of waiting</td>
</tr>
<tr>
<td></td>
<td>- Eliminate unnecessary or insufficient reviews and audits</td>
</tr>
<tr>
<td></td>
<td>- Break down silos</td>
</tr>
<tr>
<td></td>
<td>- Eliminate all non-value-adding documents, formatting, and data Handling</td>
</tr>
<tr>
<td></td>
<td>- Eliminate Unnecessary Analyses, Exploit Underutilized Analyses</td>
</tr>
</tbody>
</table>

It is recommended to do this step at a different date and time. The completed current map and identified wastes can be sent to all the participants and the participants are given one week time to think about possible strategies. Later, during a second workshop the strategies can be discussed and finalized.

### 4.4.2 S4.2: Draw future state map

**What:** In this step value stream is to be redrawn with wastes eliminated. By following the advice above, and one’s own creative solutions to the local problems, one should be able to cut most of the serious waste, particularly waiting, out of the process.

**How**:

- One of the strategies proposed for drawing current value map (see Section 4.2) can be used to draw future map. In our personal experience, workshop with all the team members works best (see Section 6.1.4).

### 4.4.3 S4.3: Re-evaluation of value*

**What:** The MIT manual does not discuss re-evaluation of value after drawing the future map which the researchers believe is an essential step during “value stream mapping”. The agreed strategies for waste reduction/elimination might be reducing/eliminating one type of waste e.g. waiting time however; they might be compromising an important value consideration e.g. quality. Thus, it is important to evaluate the impact of the agreed strategies on the derived value considerations (derived from the goal, see Section 4.1.6). The impacts could be positive and/or negative on one or several value components and these need to be evaluated because cutting costs or reducing wait times are not the only important value considerations during development.

**How:** After redrawing the current map using one of the strategies given Section 4.2, the next step is to hand over a value re-evaluation form to the participants and ask them to fill how a proposed improvement strategy (to eliminate/minimize waste) will impact all the given value considerations (selected using GQV methodology, see Section 4.1.6). For example, to achieve “efficiency”, strategies are agreed to reduce lead-time during development but it is fundamental to evaluate the effect on value consideration related to efficiency from different perspectives. For example:
1. How the agreed strategy impacts the internal efficiency value (with respect to time, quality and cost) from the company’s internal perspective?
2. How the agreed strategy impacts the delivery process value (with respect to time, quality and cost) from the customer’s perspective?
3. How the agreed strategy impacts the human capital value (competence, motivation and intellectual agility leading to innovation) from internal business perspective?

These can be evaluated using a 5-point rubric evaluation criteria as prescribed in Section 4.1.6. An illustration of how this was done in the case is explained in detail in Section 5.

4.5 S5: Perform retrospective analysis*

**What:** After drawing current state map, identifying waste, drawing a future state map and re-evaluation of value it is deemed necessary to walkthrough the future state maps and results of evaluation with the team members in a workshop. This has not been prescribed in the MIT manual but the researchers emphasize it as a fundamental activity since it allows rectifying misunderstandings during the workshop and is similar to member-checking in empirical studies (see e.g. (Robson, 2002)). Furthermore, the retrospective should include a reflection on the value stream process to identify improvement opportunities. The next time the value stream is conducted, it can be done in a more efficient way based on the feedback.

**How:** After the analysis of the current and future state map is completed, the following alternatives are available to conduct the retrospective:

1. **Informal review:** In informal review, summary of the current state map and the future state map are sent out to the participants for an informal review. No specific guide/process for conducting the review is given; it is very much up to each participant of how and when to provide the feedback.
2. **Walkthrough:** In a walk-through the person who has summarized the current state and future state maps walks through his/her report to inform the team conducting the value stream analysis about the outcome. Then the team provides feedback to rectify misunderstandings and everyone should agree on and commit to the content of the report.
3. **Inspection:** In inspection each team member inspects the report and summarizes issues found and provides feedback to the author of the report. Thereafter, a formal meeting is held where each team member reports the issues found, which are logged. Thereafter, the report is updated accordingly.
4. **Technical review:** The group reviews the document together, the process is less formal and systematic than inspections and hence it is quicker and more easily accepted by practitioners in comparison to inspections.

Further details on the methods are provided in Graham et al. (Graham et al., 2008). Table 11 provides an overview of the different techniques with respect to accuracy, time and resources, and the capability to handle complexity.
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Time and resources</th>
<th>Capability to handle complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informal Review</td>
<td>Low to medium because of high flexibility of how/when to conduct the review, only individual task</td>
<td>Low because it is individual feedback without formal requirements</td>
<td>Low to medium because of lack of exchange between different viewpoints</td>
</tr>
<tr>
<td>Walkthrough</td>
<td>High because the whole team is present at one occasion to provide feedback</td>
<td>High because all the team members have to be available at one occasion</td>
<td>High because of feedback and discussion of multiple process views in one occasion</td>
</tr>
<tr>
<td>Inspection</td>
<td>Very high because of individual preparation and the whole team being present</td>
<td>Very high because of preparation and presence required at inspection meeting</td>
<td>Very high because of feedback and discussion of multiple process views in one occasion, and well prepared team members</td>
</tr>
<tr>
<td>Technical Review</td>
<td>High because whole team is present at one occasion to provide feedback</td>
<td>High because all team members have to be available at one occasion</td>
<td>High because of feedback and discussion of multiple process views in one occasion</td>
</tr>
</tbody>
</table>

In addition, it is important to capture the feedback from the team to reflect on the actual process of the value stream, and not only the outcome. Hence, at the end each member should provide answers to the following questions:

- What was good about the value stream process (list the three main items)?
- What was not so good and can be improved (list the three main items)?
- Would you like to participate in such an improvement effort again (Rate on a scale 1-10)?

An answer of 1 implies a strong tendency towards “no”, while 10 implies a strong tendency towards “yes”. Answering with 5 indicates indifference.

5. Research Design
The research method used in this paper is case study. Case studies are particularly well suited to study phenomena in a real-world environment with the researcher making observations and consulting multiple sources of evidence (e.g. documentation, interviews, and observation) (Robson, 2002; Yin, 2003). The case being studied is Ericsson AB, a leading telecommunication company. The company is ISO 9001:2000 certified.

5.1 Case Description
The case and context are described, as this allows for generalizing the results to a specific context. Other companies in a similar context are likely to find the results transferable to their
context (Petersen and Wohlin, 2009). The process used at the company follows a system of systems (SoS) approach. There is not a common definition of system of systems, as the term has been defined in different domains, such as military, enterprise information systems, or education (Lane, 2007). The term has been recently established in the software engineering field, where a system of systems should fulfill several of the characteristics shown in Table 12. The overall architecture of the system of systems studied at Ericsson consists of 12 systems.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Case Company</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational independence</td>
<td>X</td>
</tr>
<tr>
<td>Managerial independence</td>
<td>X</td>
</tr>
<tr>
<td>Integration of system into system of Systems</td>
<td>X</td>
</tr>
<tr>
<td>SoS comprised of complex systems</td>
<td>X</td>
</tr>
<tr>
<td>System suppliers deliver systems for integration</td>
<td>X</td>
</tr>
<tr>
<td>Complete technical oversight of SoS and system supply</td>
<td>X</td>
</tr>
</tbody>
</table>

The process used at the company is shown in Figure 4. In the first step the high level requirements for the overall SoS are specified. Before the requirements are handed over to compound system development a so-called “Go” decision is taken, meaning that development resources are allocated to the high level requirements. When the decision is positive, teams specify a detailed requirements specification, which is then handed over to the concerned system(s). The requirements are then implemented for a specific development system, and they are integrated (also called system level test of last system version - LSV-test). The development is done in sprints run by agile development teams (AT Sprints in Figure 5). Each system can be integrated independently of another system, which provides them some degree of operational and managerial independence (see Table 12). However, the versions of two systems have to be compatible when the system of systems is integrated (Compound System Test). Each system is very complex, the largest system having more than 15 development teams. The size of the overall system of systems measured in lines of code (LOC) is 5,000,000 LOC. This fulfills the characteristics of SoS development related to system complexity and integration. In order to make sure that the system of systems is working together in the end an overall system structure and design is developed, referred to as the anatomy. This allows having an oversight of the overall SoS, also making explicit how each system in the SoS contributes to the overall system goals.
Looking at other context elements (Petersen and Wohlin, 2009) the following details are added:

- All systems are older than 5 years.
- On principle level the development process is incremental with projects adding increments to the code base-line on system and compound system level.
- Within the teams and in the testing activities agile practices are used, such as: continuous integration, time-boxing with sprints, face-to-face interaction (stand-up meetings, co-located teams), requirements prioritization with product backlogs, refactoring and system improvements
- The system of systems is developed globally (with some development units being located in India and some in China).

Two systems are mainly considered during the value stream activity, having a size of over 300,000 LOC each. However, due to confidentiality reasons no further details about them can be revealed.

5.2 Research question

In this case study two research questions are answered.

RQ1: How useful is “value stream mapping” to identify waste and assess improvement potential?

The answer to this question is provided by assessing the quality of the end result. That is, were the researchers able to identify issues across the development life cycle, and were the researchers able to identify improvements that have the potential in achieving the targeted goal?

RQ2: What recommendations should be given to practitioners when instantiating the framework (i.e. selecting alternatives of how to conduct the “value stream mapping” steps)?

The feedback given by the practitioners and the observations made by the researchers during the value stream activity are used as input to provide lessons learned with respect to framework instantiation.

5.3 Data Collection

Much of the data collection with respect to the “value stream mapping” activity was done and is described Section 6.1. From a case study perspective, careful observation and documentation of the process and products of the value stream activity was required. For that purpose two researchers were involved. One researcher was focused on moderating the workshop and providing guidance to the value stream map team. The other researcher was observing the “value stream mapping” activities taking notes and documenting the process (e.g. by taking photographs, taking notes and storing them in a case study protocol). The same was done by a colleague working at Ericsson, who was also providing support in the moderation process.
Photographs: The photographs document the work products produced during the workshops, such as the Zeroth maps, notes on white-boards including yellow stickers/clusters of information structured together with the team.

Notes: The notes contain as much information as possible of the discussions taking place during the value stream activity. The researcher taking the notes handed these over to the moderator in order to complement them.

5.4 Data Analysis

The data analysis was based on the photographs and the notes in order to produce the report for the value stream map activity delivered to the company. This report is also the base for answering the research questions. The notes and information from the photographs were first clustered (e.g. for the current state map everything related to a high number of parallel items was put under a cluster “number of parallel items”). The clustered information was tabulated and normatively described. The resulting report was reviewed by a follow colleague at Ericsson who was also taking notes, and took part in all activities of the value stream map. In addition, a walkthrough of the report was conducted as a means of member checking (Robson, 2002) to make sure that the information documented during the workshops is accurate.

5.5 Validity Threats

Four types of validity threats are commonly distinguished, namely construct validity, internal validity, external validity, and reliability [21].

Construct validity - obtaining the right measures for the concept being studied: The main risk in this context is that the researcher influences the outcome of the study with his/her presence. This is mainly related to trust. Given that the researcher who was moderating the workshop is also partially employed at the company, he was perceived as being internal. Hence, this threat to construct validity was reduced. Another risk is that the participants in the value stream activity misinterpret the intent and are not able to conduct the task in the best possible manner. To reduce this threat, trainings in lean and “value stream mapping” were conducted. Given that there was very limited time for training, the threat was reduced, but not mitigated. During the workshop the participants were always able to ask the moderators for clarifications with respect to the task at hand, which also contributes positively to reducing the threat of misinterpretation.

External validity/generalizability - ability to generalize the results of this study: External validity/generalizability is a common threat in single case studies, as these are only able to focus on single cases, but at the same time allow to gain an in-depth understanding of the case. The results obtained in this case study are true in the context of the case. Hence, the results can be generalized to similar context (e.g. telecommunication, systems of systems, large-scale). In order to allow others to judge the degree of generalizability researchers have taken care in describing as many contextual elements as possible using the checklist provided in (Petersen and Wohlin, 2009) as a guide. Overall, the results will be of interest to other companies in a similar context. In addition, the “value stream mapping” framework is designed providing alternatives of how to implement different activities. This makes the value stream map flexible.

Reliability - interpretation of the data is influenced by the background of the researcher/repetition of the study should lead to the same results: Whenever conducting studies with a vast amount of qualitative information, there is a risk that the interpretation of the data is influenced by background of the researcher. The risk cannot be mitigated, but only reduced. In
order to reduce this threat the following actions have been taken: (1) Two researchers were involved in the interpretation of the notes and documentation, (2) the documented result was reviewed by a colleague at the company, (3) the colleague at the company was also taking notes during the workshop allowing for a comparison of notes. In addition, member checking with the whole “value stream mapping” team has been conducted during the retrospective activity in the value stream process.

**Internal validity** was not considered a threat in this study as the researchers do not aim at establishing a statistical casual relationship between variables (this threat is mainly considered in experimental/sampling studies (Wohlin, 2000)).

6. Case Study Results

The case study results are split into two main parts. The first part illustrates how the value stream framework is instantiated in the specific case, and provides arguments of why in the given context certain decisions for instantiations were made. In the second part the actual outcome in the form of current state map, future state map, and retrospective are presented.

6.1 Instantiation of the framework

The management of the case study company agreed to commit 12 persons to the value stream map activity for approximately 10 hours (including the steps mapping the current value stream map, identifying waste, improving the process, and performing retrospective analysis), which adds up to 120 person hours. The selection of strategies and the distribution of time between steps were done under these constraints.

6.1.1 S1: Getting started

As outlined in the framework the steps for getting started are the selection of key stakeholders, determining the purpose, defining the team, training the team, bounding the problem, defining the value, and understanding value creation.

**S1.1 Key stakeholders:** Three alternative options are available, namely direct internal customers, indirect internal customers, and external customers. As the company is operating in a market-driven context where a product is developed for a mass market with many potential customers, no external customer was invited to participate. If the company would have been in a bespoke and contract based situation the external customer would be more easily accessible. Hence, only direct internal customers and indirect internal customers are included. In the case company direct internal customers are people working with the release and packaging of the software product. Indirect internal customers are participants of the process, such as people responsible for design and testing.

**S1.2 Determining the purpose:** During the planning phase the purpose was to evaluate a concrete improvement. The reason for evaluating a concrete improvement and not focusing on the overall process was to not flood the organization with too many new improvement initiatives, which would be a risk for the ongoing improvements. The concrete improvement to be evaluated was focused on re-structuring the communication between product managers, requirements engineers, and development teams in order to facilitate frequent exchange of information and feedback. This was changed after step S2.4 as the analysis showed that the improvement evaluated did not target the wastes that received the highest priorities by the team. In consequence, it was decided to change the purpose to optimize end-to-end process towards a
specific target with respect to lead-time. It was made explicit that the ongoing improvements could be incorporated when improving the current state process, but at the same time new improvements can be proposed.

The workshop participants received the following instructions for the “value stream mapping” activity: “Your task is to redraw a map that cuts the lead-time down to X days. Here you should implement improvements (e.g. the suggestions related to improved communication between product managers, requirements engineers, and development teams) and determine what changes should be done in order to achieve a shorter lead-time. Keep in mind, however, that later you will evaluate the process based on other attributes (e.g. perceived quality by the customer, cost, etc.), which might be negatively affected. Also keep in mind whether the root-causes for waste are addressed. If you cannot achieve the X days without compromising other important value considerations (see value components), determine what would be a realistic improvement.” In short, this translates into the following goal that should be achieved: “Reduce the lead-time by X% while developing usable and useful features for the customers, without defects, as efficiently as possible, and at the right time.”

S1.3 Defining the team: Who to include in the team was very much restricted by the resources committed to the value stream map. The company decided to focus on two systems (system A and B) that are part of a larger system (see description of S1.3). With the focus on the two systems, each system should have at least one representative in each process area (product portfolio, system responsibility, and system integration and release). Only chefs were included in the first value stream map initiative. The chefs were selected based on their experience in the process, i.e. they all had very good knowledge of the processes currently practiced. Also process and product owners should be present within each step. A process owner is responsible for driving process improvements. A product owner is a main stakeholder having requirements on and an interest in the product. The set-up of the team is summarized in Table 13.

<table>
<thead>
<tr>
<th>Process Area</th>
<th>Roles</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product portfolio and requirements</td>
<td>Product and system management, program manager</td>
<td>3 product owners, 1 process owner</td>
</tr>
<tr>
<td>System design, function testing, and LSV</td>
<td>Architects and designers, coders, program manager</td>
<td>2 product owners, 2 process owners</td>
</tr>
<tr>
<td>System of systems integration (C-LSV) and release</td>
<td>LSV testers, program manager, release and product packaging</td>
<td>2 product owners, 2 process owners</td>
</tr>
</tbody>
</table>

S1.4 Training the team: The teams were trained through a power-point presentation containing the following information: (1) The goals of “value stream mapping” and the potential benefits illustrated through the case study presented in the MIT Value Stream Mapping manual [4]; (2) An overview of the “value stream mapping” process; (3) ToDos related to the current state map; (4) Notations of “value stream mapping”; (5) To-Dos related to the future state map; (6) presentation of the value components with examples. After the presentation, there was a question and answer session where the content of the presentation was clarified.

S1.5 Bounding the problem: Solely focusing on documentation analysis was not considered as opinions (through discussions) are needed to understand as why specific problem bounds should be defined. Moreover, it is very important to involve key persons in the discussion who can champion the value stream activity in order to gather the required competence needed for the
initiative. Collecting as much information about the existing process before going into an interview is recommended, since the discussion can be focused on bounding the problem.

Workshops were not chosen as they require too much effort early on in the mapping process. Therefore, documentation analysis and discussion was used as a trade-offs between documentation analysis and interviews. From the documentation analysis, the researchers identified the main process steps. The following decisions were made regarding the problem bounds with respect to the critical elements defined in the framework (see description of S1.5):

1. End-to-end process: The process starts with the requirements gathering and definition activity, and ends with the system verification activity. The reason for choosing the end-to-end process is the observation that different parts of the process influence each other, and hence considering only one activity might lead to sub-optimization. For example, a change in the requirements activity could have a significant impact on testing and product packaging activities.

2. Products: The product that was chosen is a system of systems. As described in the research method (see Section 5.1) the company is employing a system of systems development approach. In total, nine systems are parts of the overall system. The most critical systems were selected for the mapping activity, i.e. a delay in the development of systems A and B has the most significant effect on the development of the overall system. This is due to the size of the systems and the number of dependencies to the other systems.

3. Product owner: For each of the process areas a process and product owner has been identified (see Table 13).

4. Customer of output: The internal customer is release and product packaging responsible for delivering the verified system and making sure it is installed and configured according to the customers’ needs. The external customers are telecommunication operators on the market.

5. Delivery of input: The high level needs are identified by market units (e.g. through market surveys, customer visits). The market units then communicate with the development units regarding the requirements.

**S1.6 Defining the value and understanding value creation**: The selection of value components is driven by the goal question value approach. The goal can be subdivided into several sub-goals:

- Develop useful and usable features
- without defects
- as efficient as possible
- at the right time.

From these corresponding questions and related value components were derived, shown in Table 14.
### Table 14 - Goals, Questions, Values

<table>
<thead>
<tr>
<th>Sub-Goal</th>
<th>Question</th>
<th>Value Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Useful and usable features</td>
<td>Which functions are really needed, and how easy are they to use?</td>
<td>Functionality, quality of delivered product</td>
</tr>
<tr>
<td>without defects</td>
<td>What is the level of quality with respect to reliability, portability, and maintainability?</td>
<td>Quality of delivered product</td>
</tr>
<tr>
<td>as efficient as possible</td>
<td>Are we developing the product efficiently, and do we have the knowledge and motivation needed to do so?</td>
<td>Internal efficiency, human capital value</td>
</tr>
<tr>
<td>at the right time</td>
<td>Are we able to deliver and install the system efficiently?</td>
<td>Delivery process value</td>
</tr>
</tbody>
</table>

An overview of the instantiation of the phase “Getting Started” can be found in Figure 5.

### Step 1: Getting Started

<table>
<thead>
<tr>
<th>Step 1.1: Getting Started</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S1.1 Key Stakeholders</strong></td>
<td>[X] Direct internal customers</td>
</tr>
<tr>
<td></td>
<td>[X] Indirect internal customers</td>
</tr>
<tr>
<td></td>
<td>[ ] External customers</td>
</tr>
<tr>
<td><strong>S1.2 Determining Purpose</strong></td>
<td>[ ] Understand and communicate process</td>
</tr>
<tr>
<td></td>
<td>[ ] Evaluate concrete improvement</td>
</tr>
<tr>
<td></td>
<td>[X] Optimize towards specific target</td>
</tr>
<tr>
<td><strong>S1.3 Defining the team</strong></td>
<td>Strategic management: Chefs [X] Doers [ ] Systems: A/B</td>
</tr>
<tr>
<td></td>
<td>Architecture and design: Chefs [X] Doers [ ] Systems: A/B</td>
</tr>
<tr>
<td></td>
<td>Implementation: Chefs [X] Doers [ ] Systems: A/B</td>
</tr>
<tr>
<td></td>
<td>Delivery and Release: Chefs [X] Doers [ ] Systems: A/B</td>
</tr>
<tr>
<td><strong>S1.4 Training the team</strong></td>
<td>One hour training session on lean principles, purpose, and value stream notation and process</td>
</tr>
<tr>
<td><strong>S1.5 Bounding the problem</strong></td>
<td>[ ] Documentation analysis</td>
</tr>
<tr>
<td></td>
<td>[ ] Interviews</td>
</tr>
<tr>
<td></td>
<td>[X] Documentation analysis and discussion</td>
</tr>
<tr>
<td></td>
<td>[ ] Workshop with team members</td>
</tr>
<tr>
<td><strong>S1.6 Defining and understanding value</strong></td>
<td>Value components considered:</td>
</tr>
<tr>
<td></td>
<td>[X] Functionality</td>
</tr>
<tr>
<td></td>
<td>[X] Quality of delivered product</td>
</tr>
<tr>
<td></td>
<td>[X] Internal Efficiency</td>
</tr>
<tr>
<td></td>
<td>[X] Delivery process value</td>
</tr>
<tr>
<td></td>
<td>[X] Human capital value</td>
</tr>
</tbody>
</table>

**Figure 5 - Instantiation of the Phase “Getting Started”**

### 6.1.2 S 2: Mapping the current value stream

As outlined in the framework the steps for creating the current state value stream map are the Zeroth map, identification of tasks and flows, data collection, evaluation of value, and understanding of iteration.

**S2.1: The Zeroth Map:** The principle tasks were retrieved from documentation and measurements, and were already well known to the researchers and the organization. Hence, this
step required little investigative effort. The main steps of the Zeroth Map were defined based on the states that requirements can enter (e.g. being in specification, being in design, being in test) and so forth. An overview of the states can be found in (Petersen et al., 2009).

**Step 2.2: Tasks and flow**: The high level activities of the Zeroth Map were not further refined for the ”value stream mapping” activity. The motivation was that (1) objective measurements were available for the steps; (2) in discussion with the practitioners the researchers came to the agreement that the analysis of the high level steps will reveal the wastes and improvement potentials with respect to value. Furthermore, the map could be further refined when needed (e.g. if the researchers feel that important wastes could not be discovered due to a too high abstraction level). Given the good understanding that the researchers already had of the main flows and tasks, documentation analysis and discussion was sufficient. In other cases where the processes are not documented and measured, more time intensive methods might be required to get a more complete picture (such as workshops and/or in-depth interviews, see Section 4.2).

**Step 2.3 - Data collection**: Often data in software engineering shows a high degree of variance, be it with regard to productivity (Kitchenham et al., 2007; Petersen) or with regard to lead-times (Petersen). Hence, few critical requirements have to be chosen that would lead to general improvements in the development process, and that can be used to retrieve measurements and set a baseline for the ”value stream mapping” activity. In this case, the requirements chosen should (1) affect multiple systems in the systems of systems structure as single system requirements are less problematic to develop; (2) the requirements should be of medium size according to the interval set by the company. Three requirements fulfilling these criteria were selected and based on these requirements an initial draft of the value stream map was prepared including actual measurements. The requirements were selected based on documentation/measurements, and discussion with the practitioners at the company. After having an initial map with tasks, flow, and measurement data a workshop was conducted to finalize the current value stream map.

S2.4 is described in S3 as these steps were combined.

**S2.5: Understanding iteration**: An approach for understanding iteration was not implemented in this value stream map since the Zeroth map was used as an input for the activity, given that measurements were available on that abstraction level. However, this does not mean that challenges related to iteration will not be discovered. In fact, several root-causes related to iterative behavior in the current value stream map were identified (see Section 6.2 – loopbacks, test cycles, rework iterations/steps).

An overview of the instantiation of the phase “Mapping the current value stream” can be found in Figure 6.

<table>
<thead>
<tr>
<th><strong>Step 2: Mapping the Current Value Stream Map</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S2.1 The Zeroth Map</strong></td>
</tr>
<tr>
<td><strong>S2.2 Tasks and flow</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
S2.3 Data collection

- [X] Documentation analysis
- [ ] Interviews
- [ ] Documentation analysis and discussion
- [X] Workshop with team members
- [ ] Follow the work

S2.4 Evaluation of value
The evaluation of value was done with S2.3

S2.5 Understanding the iteration
N/A

<table>
<thead>
<tr>
<th>Step 3: Identifying Waste</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3: Waste identification</td>
</tr>
<tr>
<td>[ ] Documentation analysis</td>
</tr>
<tr>
<td>[ ] Interviews</td>
</tr>
<tr>
<td>[ ] Documentation analysis and discussion</td>
</tr>
<tr>
<td>[X] Workshop with team members</td>
</tr>
<tr>
<td>[ ] Follow the work</td>
</tr>
</tbody>
</table>

Note: S3 was combined with S2.3 in a single workshop.

Figure 6 - Instantiation of the Phases “Mapping the current value stream” and “Identifying the waste”

6.1.3 S3: Identifying the waste

This step was combined with S2.4 (evaluation of value). The reason being that all activities contribute some value (either external or internal). For example, everyone would argue that testing is valuable. Hence, in this case the researchers found it is more suitable to start focusing on improvement potential based on long waiting/lead-times. The workshop as an approach was selected as people in different roles should interact with each other and provide immediate feedback. This could not be achieved with any other alternative provided (e.g. individual interviews, study of documentation). Furthermore, agreements and disagreements are revealed immediately and can be resolved. In the end of the current value stream map activity it is important that everyone in the team has the same understanding of the current state map and agrees on the outcome. This is important as the current state map serves as the reference for creating the future state map. In the workshop, the team was asked to perform the following activities:

1. Review data collection: Review the prepared times (lead-times, waiting times, processing times) and change them according to their experience. (30 minutes)

2. Identify wastes: Identify parts in the process with the highest improvement potential; e.g. in which phase do we see the majority of waiting, where are the activities that take very long time? This was done by putting burst signals within the value stream map containing the updated and agreed upon lead and waiting times. The burst signals were then prioritized by the team. For prioritization, cumulative voting was used as this is known to be a fast and accurate method to prioritize items (Leffingwell and Widrig, 1999). (30 minutes)

3. Identify reasons for wastes: Identify the reasons /root causes for long waiting times/processing times and prioritize the root-causes. This was achieved by having each participant write the main root-cause for each of the burst signals on a yellow sticker. The moderator (second author of the paper) collected the yellow stickers and clustered them with the help of the workshop participants. (45 minutes)

4. Information about next steps (5 minutes)

A lunch-break was introduced between workshop activities 2 and 3 in order to be able to make adjustments to the workshop if things do not go as planned. However, in this case no change in plans was necessary.
6.1.4 S4: Improve the processes

This step contains of three sub-steps: (S4.1) identify strategies to eliminate the waste; (S4.2) draw the future state map, and (S4.3) to re-evaluate value. S4.1 and S4.2 were combined in a single workshop. Other alternatives would have been to conduct document analysis, interviews, or a combination of document analysis and discussions. However, for the same reason presented in S3 (see Section 6.1.3) workshops were conducted, i.e. to allow for interaction and immediate feedback between team members. The workshop was structured as follows:

- **Introduction:** Presentation of the results from the first workshop (i.e. current state map, root-causes) was given to the participants to refresh their memory of the results. Furthermore, the value components were presented again, as these need to be considered when proposing the improvements. (30 minutes)

- **Identify improvements and redraw map:** The participants were to identify improvements that can tackle the root-causes, and to redraw the process based on the improvements identified. Ongoing improvements (if there are any) should be taken into consideration. As the number of workshop participants was 12, there was a risk in having too many people in one workshop which could lead to the likely outcome that the results could not be obtained in time. Therefore, the team was split into two sub-teams, making sure that each role is represented in each sub-team. Furthermore, before the teams were split, it was stressed that the teams should consider medium sized requirements under realistic assumptions, i.e. that there is much ongoing work in parallel, and that there is high market pressure with respect to lead-time and quality. That is, the new scenario should not be over simplified or based on “best case” assumptions. For redrawing the map the teams were given 90 minutes. When the time was up, each sub-team presented their map and a comparison /consolidation of the maps was done by agreeing on the improvements identified.

- **Re-evaluate value:** For the re-evaluation of the value each individual filled in a form answering the degree of value gain/loss based on the improvements. The scales on which value gain/loss was evaluated are presented in Appendix B. Furthermore, each team member was asked to write down why they thought the identified improvements impact value the way they answered.

An overview of the instantiation of the phase “Improve the Process” can be found in Figure 7.

<table>
<thead>
<tr>
<th>Step 4: Improve the Process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S4.1 Eliminate waste</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>S4.2 Draw future state map</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>S4.3 Revaluation of value</strong></td>
</tr>
</tbody>
</table>

*Figure 7 - Instantiation of the Phase “Improve the Process”*
6.1.5 S5: Retrospective

The retrospective was conducted in two steps. In the first step a walk-through of the results was done. In order to conduct the walk-through the moderators prepared a written report describing the results obtained in the previous steps. Walk-throughs are particularly well suited to inform a group of people about the result, and to collect feedback. During the walk-through every participant received a printed version of the report, and the moderator went through the report page by page explaining the contents of each page. The participants were allowed to interrupt and ask questions at any time of the walk-through.

In the second step, the researchers collected feedback on the “value stream mapping” process asking three questions:

- What was good about the process?
- What was not so good and how can it be improved?
- Would you like to use “value stream mapping” as a tool for improvement in the future? Rate on a scale from 1 (no, never again) to 10 (should be a continuously used method for improvement). A value of 5 means you are indifferent.

An overview of the instantiation of the phase “Perform retrospective” can be found in Figure 8.

<table>
<thead>
<tr>
<th>S5 Retrospective</th>
<th>[ ] Informal review</th>
<th>[X] Walkthrough</th>
<th>[ ] Technical review</th>
<th>[ ] Inspection</th>
</tr>
</thead>
</table>

**Figure 8 - Instantiation of the Phase “Perform retrospective”**

6.2 Outcome of framework application

Three main work products were produced during the value stream activity:

1. Current state map including Zeroth map, burst signals indicating improvement potential due to undesired behavior, and root-causes for the burst signals.

2. Future state map including improvement proposals, a future state map indicating changes based on improvement proposals, and estimates for new lead and waiting times. Furthermore, the re-evaluation of value was conducted.

3. Retrospective containing reflections on the value stream process and the delivered result. In the following a narrative of the work products and the reflections provided by the workshop participants are given.

6.2.1 Current State Map

**Review data collection/Zeroth map**: The workshop started with a presentation of the Zeroth Map that was drafted by the researchers prior to the workshop. The map is shown in Figure 9. It contains the following activities:

- **HL-Req**: The requirement is specified such that the development organization has an idea of the customer’s needs, and estimations for time can be made. This activity maps to High Level Specification (Prio) in Figure 4.

- **Commit for dev**: At this point a decision is made whether the development organization should commit development resources in order to implement the requirement. The
answer has to be “yes” in order for the requirement to enter the “Compound System Development” box (in Figure 4).

- **Write RS**: The detailed requirement is specified, so that it is precise enough to be developed in system development, this activity maps to the DRS activity in Figure 4.
- **Design & FT**: The requirement is implemented in system development and tested within the teams (Agile Team Sprint in the Figure 4).
- **LSV/C-LSV**: The integrated result of the system development is tested at the system level test (LSV). The compound system is tested in the compound system test (C-LSV).

![Figure 9 – Zeroth Map Current State](image)

During the workshop the team filled in the lead-times and waiting times within the activity in percentages, as these were not available as data prior to the workshop. Furthermore, some of the waiting times in-between activities were moved into the activities, as the participants argued that there is some overlap between the phases, and that items generally do not get stuck in between the activities. Given the description of the activities after 37% of the lead-time, development resources are committed to development. After 82% of the lead-time, the first verifiable version of the requirement is completed, and after 100% of the lead-time, a verified and usable feature is ready to be made available to the customer.

*Identify wastes*: Based on the lead-time/waiting time distribution the burst signals were placed, indicating improvement potential/need for improvement in terms of overall process lead-time or waiting time. The activity of drawing the Zeroth map was finalized after everyone agreed to continue with the values on the white board. In order to focus the improvement effort the bursts were prioritized using cumulative voting. The overall ranking was obtained by adding up points of individual participants for each phase. The ranking is shown in Table 15.

<table>
<thead>
<tr>
<th>Alarm</th>
<th>Description</th>
<th>Rank (Votes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL-Req.</td>
<td>Overall long time with 50% waiting</td>
<td>1 (41 votes)</td>
</tr>
<tr>
<td>Commit for Dev.</td>
<td>Mainly waiting</td>
<td>2 (30 votes)</td>
</tr>
<tr>
<td>LSV/C-LSV</td>
<td>Mainly waiting</td>
<td>3 (22 votes)</td>
</tr>
<tr>
<td>Write RS</td>
<td>Overall long time with 50% waiting</td>
<td>4 (16 votes)</td>
</tr>
</tbody>
</table>
Identify reasons for waste: Based on the clustering of the issues in the workshop a matrix was obtained showing general issues and their mapping to the different process activities. The root-causes helped the participants to target the right issues when searching for improvements. As can be seen from Table 15 the high level requirement specification (“HL-Req.”) have the highest improvement potential, based on the opinions of team members, followed by “commit for development”, “LSV/C-LSV”, and “Write RS”. Table 16 illustrates the result obtained from the workshop. What should be observed here is that a number of general causes have been identified (such as many parallel items). Some of the general causes apply to several phases while others are unique to a specific phase. The causes are not elaborated in further detail, the purpose of the table is to provide a general overview of what type of causes have been identified to demonstrate the capability of the value stream approach to lead to a good coverage of causes across the life-cycle.

Table 16 - Root causes vs. Activities

<table>
<thead>
<tr>
<th>General cause</th>
<th>HL-Req</th>
<th>Commit for Dev.</th>
<th>LSV/C-LSV</th>
<th>Write RS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many parallel items</td>
<td>many HL-req. in development in parallel; too high ambition in the beginning, ending up with less in the end</td>
<td>x</td>
<td>too many deliveries shortly before release intent and late system integration</td>
<td>x</td>
</tr>
<tr>
<td>Level of detail</td>
<td>High quality early for good estimates, too high ambition (document up to 80 pages)</td>
<td>Discuss too much detail delaying decisions</td>
<td>x</td>
<td>specify solutions, not requirements; different expectations on level of detail</td>
</tr>
<tr>
<td>Involve too many/wrong parties /communication</td>
<td>estimations/solutions require many people; document/feedback oriented negotiation</td>
<td>Waiting for clarifications</td>
<td>x</td>
<td>Lack of communication/cooperation between requirements, implementation, test; long time for handovers</td>
</tr>
<tr>
<td>Change (ability to change, request of change)</td>
<td>new input from customer</td>
<td>waiting to allow for second thoughts as desired outcome not clear enough</td>
<td>x</td>
<td>come up with new ideas/solutions due to late involvement of design</td>
</tr>
<tr>
<td>Size</td>
<td>x</td>
<td>too large requirements to fit into release scope (cannot approve parts of requirements)</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Release focus</td>
<td>x</td>
<td>try to fit requirements</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>General cause</td>
<td>HL-Req</td>
<td>Commit for Dev.</td>
<td>LSV/C-LSV</td>
<td>Write RS</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>-----------------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>into certain release instead of starting to work early, cannot commit resources on parts of requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loopbacks</td>
<td>x</td>
<td>x</td>
<td>repeated fix/correct cycle for internal faults</td>
<td>x</td>
</tr>
<tr>
<td>Length of test cycles</td>
<td>x</td>
<td>x</td>
<td>waiting for final delivery before staring of test; too many test cycles</td>
<td>x</td>
</tr>
<tr>
<td>Many re-work steps</td>
<td>x</td>
<td>x</td>
<td>long time due to many activities - understanding problem, generating new package, re-verifying</td>
<td>x</td>
</tr>
<tr>
<td>Quality of tests / test efficiency</td>
<td>x</td>
<td>x</td>
<td>Too big fault-slip from systems to compound system integration</td>
<td>x</td>
</tr>
<tr>
<td>System-focus</td>
<td>x</td>
<td>x</td>
<td>System focus performs well, challenges in integration performance at compound level</td>
<td>x</td>
</tr>
<tr>
<td>KPI influenced process</td>
<td>x</td>
<td>Key performance indicators drive wrong behavior (wait too long with starting in development)</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

6.2.2 6.2.2. Future State Map

To draw future state map the team was again split into two groups. Each group returned with very similar improvement ideas, and easily agreed on a combined list of improvements. The groups only differed with respect to visualization of the results. The improvement ideas are listed below.
1. The level of detail in which the high level requirement is presented should be drastically reduced. The high level requirement should be one-pager, which should be used as an input to define user stories (i.e. a high level requirement is sliced into several high level user stories - from here on called epic user stories). The improvement addresses the cause of “level of detail” (see Table 16).

2. Integrate value (epic user stories) early: One of the challenges identified was that parts of a high level requirement could not be integrated and verified early on. This was partially due to the way requirements were committed to the development organization (i.e. not parts of a high level requirement could be committed) and the release focus (see Table 16). As a solution a high level requirement should be split into epic user stories that are (1) verifiable and (2) deliverable. That is, when defining the requirements it is important to find a good modularization in terms of features for each high level requirement. An epic user story then can be verified more independently and be useful to the customer, even when some epic user stories related to the requirement are not yet implemented.

3. Communication and interaction: People from different roles in the organization should be put together to allow for better communication and understanding. For example, persons responsible for system management and development teams should interact and there should be frequent interactions so that system managers are aware of what is happening in the team developing their requirements. Also, it is important to assure that people across the overall development life cycle interact, which means involving release and product packaging teams earlier and merging system managers, program managers and development teams to communicate frequently (e.g. by co-locating, assure frequent possibilities for interactions).

4. Hand-over and documentation: The amount of hand-overs between system management and development in parallel should be reduced to increase direct communication and interaction (face-to-face interaction and communication replaces/reduces/simplifies documentation).

5. Increase parallelization: Parallelization between systems and within a system should be increased (e.g. in order to avoid systems waiting for each other). However, it is not clear how to achieve this. One of the challenges is to allow for backward comparability to always be able to integrate with another system in the systems of systems structure. For example, if system A is updated to a latest version, and system B is not updated yet, a system integration cannot be run due to lack of compatibility (as system A is not backward compatible).

6. Inspection/Review: Inspections and reviews of each other’s documentation should be reduced and direct communication across roles should be used instead (related to bullets 2 and 3).

7. Welcome change of plans: Resources for epic user stories should be reserved in plans, and the progress should be continuously reflected upon. Based on the reflection flexibility is needed when it comes to re-planning to get a higher return on investment (re-planning for value).

8. Retrospective: Continuous reflection on work processes and plans should be done to learn and improve the ways of working.

9. Team interaction: Team members of different systems should be combined to achieve a system/solution perspective of the overall system of systems instead of just having a single system perspective.
10. Prototyping: When useful, prototyping should be done to achieve a better understanding of what is needed by the customers.

What set the teams apart was that one team introduced prototyping as a solution. Otherwise, both teams were in-line with their improvement proposals. However, when it comes to visualizing the solutions both teams have drawn different pictures. The pictures have some notable differences, but at the same time share the ideas of the improvements listed before.

**Result of Group 1:** The new process map by group 1 is shown in Figure 1. The left upper box represents the new high level requirements document (QSM 1 pager) that is reduced in complexity. There is an estimated waiting time in between the completion of the QSM 1 pager and the start of development. The pre-requisite for starting is to take a so-called “Go” decision on an epic user story, indicating that resources of the development organization should be committed to the implementation of the epic user story. When this is done, the epic user story is specified and broken down in development user stories. These are then developed and function tested, and thereafter system tested and compound system tested. According to the team system testing and compound system testing can be parallelized to a high degree. What is also visible is that activities are ongoing in parallel. For example, while the third epic user story is specified, the user story from the second epic is specified, and the user story of the first epic is implemented and function tested. Looking at the time-lines the following improvements are estimated, as shown in Table 17. The table shows the % of lead-time needed to complete certain goals. The first improvement is related to when the organization is able to commit the resources for a high level requirement of medium size. The next row shows when the first verifiable unit to test is delivered. In the current state map this was at 82% of the overall lead-time, while in the new process it is only 22.50% of the overall lead-time needed in the current state map. The reason is that the high level requirement is now split into verifiable sub-units (called epic US slice). Hence, the first verified and deliverable unit is ready after 38.10% of the old lead-time, while the first verifiable and delivered unit in the old process could only be ready at 100.00% of the lead-time. To complete the same amount of functionality (the estimate of the team was that a medium sized high level requirement could be broken down to approximately 5 epic user stories) 50% of the old lead-time would be needed, given the high degree of parallelism. With regard to waiting time the estimations were extremely optimistic, going down to 2.25% from 47.40%.

<table>
<thead>
<tr>
<th>Lead-time</th>
<th>Current</th>
<th>Future</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to commit resources</td>
<td>23.00</td>
<td>9.00</td>
<td>14.00</td>
</tr>
<tr>
<td>Time to implement first verifiable unit for test</td>
<td>82.00</td>
<td>22.50</td>
<td>59.50</td>
</tr>
<tr>
<td>Time to complete test of first verifiable /deliverable unit</td>
<td>100.00</td>
<td>38.10</td>
<td>61.90</td>
</tr>
<tr>
<td>Time to complete the same amount of functionality</td>
<td>100.00</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>Ratio waiting time/productive time</td>
<td>47.40</td>
<td>2.25</td>
<td>45.15</td>
</tr>
</tbody>
</table>

**Result of Group 2:** The results of Group 2 look very similar to the results of Group 1, they are just expressed in a different way (see Figure 11). Both maps show parallelism with regard to requirements work, implementation, and testing. The difference is that the Group 2 has expressed the overlap in roles in the visualization. Furthermore, prototyping as a concept has been stressed.
This group also stressed that value should be created continuously. Thereby they separated value in terms of function and quality, according to the definition of value components in Appendix B. The value growth of the current in comparison to the future state map is illustrated in Figure 12. The figure shows that currently, features are integrated quite late in the development process, and quality assurance has no other option than following function integration in time. This is very much in line with the observation in the current state map, where the first verifiable unit is delivered to test at 82.00% of the lead-time. In the new state with the improved process, features are integrated continuously, which also allows for continuous quality assurance.

Figure 10 - Future State Map Group 1
Furthermore, in the new situation the curve is not perfectly linear, as illustrated in the Figure 12. So far, the re-evaluation has mainly been focused on time, but the participants also evaluated the impact of the set of improvements on the value components.
The evaluation of the participants with regard to the value component “External Customer Value” is shown in Figure 13. The figure shows a high agreement that functionality and quality are positively influenced, however, there is some doubt with regard to the effect on external quality.

Positive Effect: For the positive effect on functionality the participants provided the following reasons. Firstly, the new approach increases flexibility by splitting up requirements into epic user stories that are deliverable and verifiable (i.e. requirements are split based on value considerations). The epic user stories also allow for earlier feedback of whether the right scope is delivered. Functionality is delivered more accurately as the company is not implementing what has been decided a long time ago, as (given the dynamic market) this would not be what the customers want today. Also, the issue of parallel items is addressed as not everything is done at once, and only parts of the functionality can be delivered. No negative effects have been identified for functionality. With respect to quality positive effects have been identified. Due to more frequent deliveries quality is assured more often. Furthermore, the reliability is said to be improved due to shorter feedback cycles and hence smaller changes have to be delivered to test. Fewer defects being reported will also increase maintainability. As can be seen in Figure 13 four persons think that there is no positive effect on quality with respect to usability. Their reasoning was that usability is not affected as improving usability requires a higher involvement of the real customer (e.g. through participatory design, and creating mockups with the customer). Furthermore, the requirements elicitation should move more from a function centric to a quality centric dialog with the customers.

The evaluation of the participants with regard to the value component “Internal Efficiency” is shown in Figure 14. Disagreement is visible regarding the impact on cost.

Positive: Most people agree on positive impact on time and quality, however, there is some doubt that there is an effect on time, cost, and quality. Furthermore, three persons saw a negative effect on cost. The positive effect on time was explained by having higher parallelization right from the start, meaning not so much time spent on guessing but rather doing and delivering, and that the focus on demos and working software will provide results earlier. With respect to cost few persons suspected positive effects, but did not provide an explanation. The positive effect on quality of work products follows the same argumentation as for external quality.

No effect: The figure shows that several participants said that there is no effect on time/cost/quality. For time, they suspected that it will probably the same, the only difference being the content is delivered much more accurately (i.e. they felt that the estimations were over-
optimistic). They also had the feeling that there will be changes in several areas, but overall the time might be the same. For cost, some suspected that the cost is just moved from one point to another, which some also believed is true for software quality.

Negative: People suspecting a negative effect on cost mainly explained this with the increased cost of cross-functionality, as more resources have to be involved at the same time, such as system management has to support people doing implementation, which would require more people filling in different roles.

The evaluation of the participants with regard to the value component “Delivery Process Efficiency” is shown in Figure 15. There is disagreement with regard to the effect on the delivery process; overall the majority thinks that there is no effect.

Positive Effect: A positive effect in quality can be seen since by frequent installations and roll-outs; the company can learn and get better in delivering upgrades. Also, the quality in the delivery process is improved with product quality. A positive effect on time was argued by saying that the releases would be delivered faster if product packaging and release are involved earlier, as then they can prepare in parallel to development. Furthermore, partial deliveries will be possible with the new process. With respect to cost, a positive effect is seen because better quality of the system will ease roll-out and installation during deliveries.

No effect: On the other hand, people saying there is no effect on time argue that it still takes time to get the large systems running at the customer site given that each customer has to re-test in their own environments. Furthermore, when not going away from a focus of few major release deadlines per year there will be no change as homework behavior can be observed, i.e. work is done more intensively closer to deadlines. In other words, if we ought to deliver continuously work will be done more continuously over time. The reason of few major releases was also raised with respect to no change in cost as the size and complexity of the product delivered to product packaging would still be the same. No effect on quality was seen due to that there is still a focus on major releases, which is not seen as a good driver for frequent verification/delivery of intermediate value.

Negative effect: The person stating that the changes have a negative effect was saying that it is a gut feeling, not having a good explanation for the reasons.
The evaluation of the participants with regard to the value component “Human Capital Value” is shown in Figure 16. The figure shows that there is a high level of agreement on strong positive effects on competence, learning, intellectual agility.

Positive effect: The reason reported for increased competence and intellectual agility was that cross-functional work teams will contribute positively to competence development. However, the participants suspected that the competence would change to having a broader knowledge, but not necessarily deeper knowledge in a specific area. For increase in motivation, many different reasons have been reported. For example, people will get more motivated by knowing the reason of why they are doing a specific task, the reasons being known due to interactions between different roles. That is, a designer and a tester will see a market value of what he/she is producing. The motivation also increases when everyone can see the whole chain/value flow. Early feedback and understanding of features and functionality will also be a motivator. Finally, the new suggested way of working will challenge employees in a new way (e.g. with respect to formulating high level requirements in one page and then communicate instead) and hence will make working more fun. The summary of the results of the value re-evaluation shows that the survey allowed getting a balanced view of the impact of the improvements on value.
6.2.3 Retrospective

During the retrospective the moderator walked the participants through the report. Afterwards, an open discussion followed. The discussion was based on the way forward. That is, what obstacles and open questions are there today in order to achieve the new process, and what can be done to remove the obstacles and answer the questions. Based on the discussion four key questions have to be answered:

1. How do we change our ways of estimating and planning?
2. Which variable level of detail should we aim for?
3. How do we synchronize in a good way, and what is a reasonable level of synchronization/unsynchronization?
4. How do we align our efforts with development taking place in India?

For the questions the following answers have been provided:

1. Planning was done based on major release intent. However, in the new situation where more flexibility is possible and not so much details are available planning has to change as well (e.g. how to handle estimations). This also raises the question as how to handle release planning and road-mapping of features? A solution for this question has not been provided at this stage.

2. This question is related on how to get development started within 9% of the lead-time being spent on high level requirements. There might be different needs with respect to detail (some might need more detail, while others need less). One possible solution was provided by looking at company internal success stories. Experiences from one product showed good results with prototyping, which means that full picture is not needed before starting. Moreover, requirements stated by product managers in an understandable and precise manner also help to start working. Currently, requirements are often one-liners, but there are good examples from other systems recently as how to write high level requirements in an understandable way. A way forward might be to find these “model”
requirements as a guide for improvements (e.g. good use cases). Complementary, workshops with software product managers and developers early on can also be useful.

3. In order to achieve shorter lead-times in the early phase (before the decision of committing development resources), systems need to be able to work more out-of-sync. However, too much out-of-sync causes a risk when requirements change and the systems do not work as a whole after integration. A possible solution is that one system takes lead in developing the core and other systems catch up building on the core. Furthermore, the degree of dependencies between nodes highly depends on the handling of interfaces and network protocols in order to allow for backward compatibility.

4. Improvement efforts should be aligned with improvement efforts done in India. A possible solution proposed was to conduct a similar value stream activity with development sites in India to identify their idea of process improvements focusing on value and waste. The proposition/gut feeling is that when drawing the future state map India would like to have the technical details in place before starting.

After having discussed the way forward, the participants assessed the overall value stream activity saying what went well, what can be improved, and whether they would like to have "value stream mapping" implemented as a process improvement method. The following points have been raised as being positive throughout the value stream activity:

- Very good discussions in the workshop.
- Every part of the organization was involved to exchange views and learned from each other.
- Good reporting/documentation of the value stream activity.
- The activity helped everyone to understand the way of working/processes end-to-end.
- Pre-meetings and trainings were good to mitigate the risk of not discussing the issues at hand in the workshops.
- Everyone participated in the workshops after being invited.
- The view and consideration of different values.
- The end-to-end focus.

The following points have been raised as being not so good throughout the value stream activity:

- Only chefs, not doers in the workshop.
- The “value stream mapping” used in the start was not a good representation (did not show overlaps between activities), the requirements were not a common case, the most complicated case was selected, but these shortcomings were overcome in the workshops.
- The process is perceived as being too academic.
- Time intensive to conduct (but, as commented: if actions follow, very much worth it).
- The solutions were quite black and white and oversimplified.

Three improvement proposals have been provided:
• Shorter time and fewer people vs. involve India and also include technical people.
• Clearer expectations on the outcome.
• Present goals of the workshops more clearly to everyone.

The final step in the “value stream mapping” activity was to capture the attitude of the team towards the overall activity. The results are shown in Table 18. The table shows that the majority thinks the workshop was very useful. Only a few doubted the usefulness. Observe that not all persons (9 out of 12) provided a vote, as they would make their vote dependent on the support in implementing all the proposals made. In other words, they believe that the improvements identified have a positive effect, but their identification is only worth when they are implemented. A comment from one of the participants was that if the proposed improvements are implemented, the workshop would get a 10, otherwise a 0.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Number of persons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workshops very useful (&gt; 8 points on scale)</td>
<td>5</td>
</tr>
<tr>
<td>Workshops useful (Score 5-7)</td>
<td>2</td>
</tr>
<tr>
<td>Workshops not so useful (Score&lt;5)</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 18 - Attitude Towards Value Stream Mapping Activity

Given the results presented in the case study, the researchers provide a number of recommendations/lessons learned, also referring to the evidence based on which the recommendations are provided.

7. Discussion and Lessons Learned

In the following a number of lessons learned from conducting the “value stream mapping” activity are presented. For each lesson the researchers refer to the evidence collected in the case study that supports the lesson.

Use workshops for waste identification (S3), waste elimination (S4.1), and drawing the future state map (S4.3). In the retrospective, fruitful discussions in the workshops were appreciated. Direct interaction was possible, in fact people from different parts of the organization (e.g. release and product packaging) hear people talking about preceding process steps that are of interest for their work, but they have not been able to meet the persons before in order to learn about their process and communicate the knowledge. Given the context of the study (systems of systems) with many teams and activities, it is important that people are given the possibility to meet and discuss process issues who otherwise do not meet in the daily work, also recognizing that they need to communicate in the future. This was also apparent from the root causes being identified, many being related to sharing knowledge in forms of documentation and communication. Also, improvements proposed stressed face-to-face communication across roles and the preference of direct communication over documentation and review.

In smaller organizations where people work in small/few teams other approaches might be preferred (e.g. interviews) given that overall fewer resources are available, and everyone knows the overall process/project structure well. In order to investigate this, further instantiations of the framework are needed.
**Assure cross-functionality and end-to-end focus.** The end-to-end focus is generally stressed in the lean methodology (cf. (Petersen and Wohlin, 2010, 2011; Poppendieck and Poppendieck, 2008)). As can be seen in the results, root-causes have been identified for every stage of the process, which is often not possible when only looking at one activity. For example, an undesired behavior such as long waiting in a specific phase does not necessarily have the root cause in the same phase. With respect to improvement proposals the process was redrawn end-to-end, which also required the input from all parties during the discussion. An example of this is the discussion of obstacles in order to implement and make the newly proposed process changes work (see Section 6.2.3). In order to do highly parallelized development work the synchronization between teams is technically challenging, which could have been overlooked when only looking from the requirements perspective (as the requirements engineers lack knowledge about the technical dependencies in the solution architecture).

**Time-box activities in workshops when having very limited time, and combine value stream steps.** As stated in the beginning of Section 6.1.1 very limited time was available for the value stream activity (in total 10 hours). Such situations are likely to occur in practice when a method is piloted and not yet established as an accepted practice. With limited time the researchers found that it is essential to time-box activities in the workshops, the time for the time-boxes being stated for each activity in the instantiation in Section 6.1. Thereby, it is important to inform the participants about the time-boxes and to make sure that they are kept. The time-boxes should be planned with buffers (e.g. add 10 minutes on each time box) to allow for resolving eventual issues that come up during discussions and require more time. Given the restricted time, some steps were also combined in one workshop. However, the researchers believe that it is important to have time between drawing the current state map and the future state map in order to allow for reflection about the workshop related to the current state map.

**Stress complexity and realistic context during the whole activity.** In the retrospective (Section 6.2.3) an issue was raised that solutions are black and white and oversimplified. The reason is that while creating the new map an optimal context seemed to be assumed by the participants (i.e. few dependencies to other systems, ability to plan). Hence, it is always important to stress a realistic context. Therefore, the researchers propose to prepare scenarios for the participants that should be evaluated against the new future state map (e.g. what if a project is not completed in a sprint, what if there are many dependencies to another system and that system is delayed, and so forth). These scenarios could be used to update estimation values. The oversimplification issue has been partially addressed in this value stream activity already in the reassessment of value and the retrospective (Section 6.2.3). In the re-evaluation of value the participants provided possible risks why the improvements might not have an effect, and in the retrospective they discussed the main obstacles/issues that have to be resolved in order to make the improved process work. The researchers believe this provides important insights in order to succeed in implementing a version of the future process that is close to the one proposed in the future state map.

**Assure handshaking of goals and expected outcomes as early as possible.** From experience with previously held workshops in industry the researchers know when the goals of an activity are not clear, they are discussed during the workshop, consequently the actual task is not completed. In order to focus the effort on the actual value stream activity the goal of the “value stream mapping” activity was hand-shaked with the management and each participant received written documentation of the goals. Given that the activity was successfully completed within the time-boxes, there is a strong indication that the initial effort in defining and agreeing on the goals and expected outcomes of the activity contributed to a complete result from the “value stream mapping” activity, including current state map including root-causes, future state map and
improvement lists, as well as revaluation of value components. That is, if the researchers had not hand-shaked the goals and outcomes before (i.e. everyone agrees to them) these could have been discussed continuously without focusing on the remaining value stream activity tasks (such as root-cause analysis, improvement identification, etc.).

**Document the process and products produced during the “value stream mapping” activity carefully.** In the retrospective the participants appreciated the quality of the reporting (Section 6.2.3). It is central to capture and document the activities carefully as a vast amount of information is produced with respect to root-causes, improvements, process charts, reflections, and so forth. To be able to later on take actions and spread the knowledge of the results in a very large organization, careful documentation and analysis is the key. Hence, during the workshops a scribe should be present, as the moderator would not be able to capture the discussions taking place, and at the same time ensuring that all desired outcomes are achieved. In addition to the detailed report, short information material should be prepared that could be presented at department/project/team meetings in order to easily spread the knowledge. The report can then be used as a reference to learn about the detailed results.

**Allow teams to express processes in their own words.** During the initiation, there was a question whether to require a specific notation for the new process, or to let people express the process in their own words. The researchers decided for the latter alternative as participants indicated it would easier to express the ideas and improvements in their own ways. However, the only requirement posed was the times (waiting as well as processing times) could be derived from the new process map, allowing for comparability to the current state map. As is indicated in the two maps produced by the groups, a comparison could be made for a high level requirement of similar size and system impact. Hence, there is no foreseeable reason to prescribe a specific notation. In short, it is important what information is captured, and not how. In addition, the maps should be complemented by information regarding discussions that the scribe should document (see previous lesson).

**Introduce different value components to avoid sub-optimization.** In lean development “seeing the whole” is an important principle to avoid sub-optimization of specific aspects of the software process (Petersen and Wohlin, 2010; Poppendieck and Poppendieck, 2008). In the case study reported, sub-optimization was avoided in two different ways: (1) the goal of the “value stream mapping” activity was focused on reducing lead-time, but it was given as a restriction that other values (derived from the goals) should not be compromised. In case there is a need for compromise, a realistic new goal for lead-time reduction should be set. (2) At the end of the value stream activity the improvements were evaluated with respect to multiple value components. As has been revealed in the data, not only the positive aspects were discovered. At the same time risks associated to the improvements were identified, such as an increase in cost (see “Value Revaluation” in Section 6.2.3). Hence, multiple value considerations should always be considered when conducting “value stream mapping” to make sure that no important value is compromised through improvements. In addition, the researchers recommend capturing the rationales for value re-valuation.

**Involves chefs and doers.** As discussed earlier, the number of participants in the workshop needs to be limited in order to be able to finish discussions in a reasonable time frame. In the ideal situation, everyone should be involved. The researchers completely agree with the observations of the participants that it is important to involve the doers as well. However, in this case this was not possible, since time the effort spent on the activity was already perceived as high (see “Retrospective” in Section 6.2.3). Hence, in the future it is important to conduct subsequent “value stream mapping” activities with doers, as well as with stakeholders in India.
8. Conclusion

This paper presents a framework for conducting “value stream mapping” in software engineering. The framework outlines the process of the “value stream mapping” activity, and provides alternative strategies of how to conduct each step. A case study has been conducted showing an instantiation of the framework. Based on the application answers to the research questions are posed:

*RQ1: How useful is “value stream mapping” to identify waste and assess improvement potential?*

Given the constraints for the value stream activity (10 hours with 12 persons) the framework provided support in order to complete all activities related to the value stream process. A current stream map and related waste was identified. For the waste a total of 23 root causes were identified across the overall life-cycle. Based on the identified wastes and the root-causes improvements were identified. The improvement suggestions were evaluated by the practitioners and led to predicted improvements of 50% overall lead-time reduction. From inception to time to implement a first verifiable user story the time was reduced by 59.5% and the time to complete test of the first verifiable/deliverable unit was reduced by 61.9%. The reductions were based on the assessment in the value stream map activity. In addition, the re-evaluation of value showed that the suggested improvements mostly have a positive impact on external, internal, human capital, and delivery value. Where this was not the case, obstacles were identified to be solved in order to achieve the improved process. Overall, a retrospective showed that the “value stream mapping” activity was perceived as valuable, and that people would support having it as an established method for improvement in the organization. Given the limited time available, the activity has provided rich results about the current process, and opened up new ways forward of significantly improving the current process.

*RQ2: What recommendations should be given to practitioners when instantiating the framework (i.e. selecting alternatives of how to conduct the “value stream mapping” steps)?*

Nine lessons learned have been presented in the paper. They are:

1. Use workshops for waste identification, waste elimination, and drawing the future state map to allow for exchange and discussions between team members;
2. Assure cross-functionality by bringing together roles and involve people from the whole value chain (end to end);
3. Time-box activities in workshops when having very limited time, and also combine different “value stream mapping” steps on single occasions;
4. Stress the complexity and realistic context during the whole activity by throwing in scenarios and how these scenarios would impact the new process in terms of waste and value;
5. Assure handshaking of goals and expected outcomes as early as possible to avoid risks of not completing the “value stream mapping” task with given resources;
6. Document the process and products carefully to support implementation of suggested improvements;
allow teams to express the improved processes in their own way, but assure comparability to the current state map;

(8) introduce different value components to avoid sub-optimization (e.g. improving lead-time and compromising other components, such as quality);

(9) involve chefs and doers when possible.

As a future work, plans are to instantiate the “value stream mapping” framework in different contexts (such as different domains, different system complexities and structures, etc.).

9. References
Chase, J.P., 2001. Value creation in the product development process, Massachusetts Institute of Technology, Massachusetts
Khurum, M., Gorschek, T., Wilson, M., 2011. Software Value Map – An Exhaustive Collection of Value Aspects for the Development of Software Intensive Products. Submitted to a journal, can be obtained from authors upon request.
Rother, M., Shook, J., 2003. Learning to see: value stream mapping to create value and eliminate muda. Lean Enterprise Institute, USA.
Appendix A - Evaluation of Strategies with Respect to Accuracy, Time and resources, and Capability to handle complexity

Different strategies were evaluated based on the attributes of accuracy, time and resources, and capability to handle complexity. In the following a definition of the attributes is provided.

**Accuracy**: Accuracy is defined as the ability of the method to capture information on the process in a way that it reflects reality. For example, the information about a process elicited in an interview can be very subjective and not in line with how the process is actually executed. Four levels of rating accuracy have been used in the paper, which are defined as follows:

- **Very high**: The data collected reflects the reality and is complete with respect to the information need (e.g. enough information is available to identify waste).
- **High**: The data collected reflects the reality and is partially complete with respect to the information need.
- **Medium**: The data collected is likely to contain subjective information and is partially complete with respect to the information need.
- **Low**: The data is likely to contradict reality and is incomplete.

**Time and resources**: Time and resources determine the effort to conduct the different steps of the value stream map. For example, if in a workshop many keypersons have to be tied up at once for a longer period of time (e.g. full day) would be rated as higher effort than studying documentation. Four levels of rating time and resources have been used in the paper, which are defined as follows:

- **Very high**: All members of the value stream team need to prepare individually in advance and are tied up for a longer period of time all together at one occasion (e.g. half-day work-shop with everyone).
- **High**: No preparation is required, but all team members are tied up for a longer period of time all together at one occasion (e.g. half-day work-shop with everyone).
- **Medium**: Individual members of the team have to allocate time, but necessarily in the same time period.
- **Low**: None of the team members have to allocate time.

Note that the time of the organizer of the value stream map (e.g. moderator of the workshops) is not considered here.

**Capability to handle complexity**: The capability of the strategy to handle complexity refers to the ability of capturing relationships between different parts of the process from an end-to-end perspective (e.g. capturing the effect of poor requirements on the remaining phases of development). Four levels of rating time and resources have been used in the paper, which are defined as follows:

- **Very high**: The overall process end to end (according to the boundary definition) is captured, and the relationships between different sub-processes are expressed.
- **High**: The overall process end to end (according to the boundary definition) is captured, and the relationships between different sub-processes are partially expressed.
Medium: A set of sub processes is well understood, but it is likely that important links/effects between them are not captured.

Low: Only individual sub processes are captured without focusing on the links/effects between them.

Note that the time of the organizer of the value stream map (e.g. moderator of the workshops) is not considered here.
Appendix B - Value Components for Value Stream Mapping

Four types of value components have been identified from the value map. These are functionality, quality of delivered product, internal efficiency, delivery process value, and human capital value. Each value component is evaluated against the improvements. The effect is rated on a scale. The following tables provide an overview of the survey being filled in by the practitioners.

B.1. Value Component: Functionality
Perspective: External (customer).
Description: The capability of the software product to provide functions, which meet stated and implied needs when the software is used under specified conditions.

<table>
<thead>
<tr>
<th>Impact</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>The improvement is critical in improving to provide functions needed by the customer (i.e. to deliver the full scope of committed)</td>
</tr>
<tr>
<td>+1</td>
<td>The improvement is important in improving to provide functions needed by the customer</td>
</tr>
<tr>
<td>0</td>
<td>The improvement has no effect on the scope.</td>
</tr>
<tr>
<td>-1</td>
<td>The improvement might slightly hinder in delivering the scope.</td>
</tr>
<tr>
<td>-2</td>
<td>The improvement is a big obstacle in delivering the scope.</td>
</tr>
</tbody>
</table>

B.2. Value Component: Quality of the delivered product
Perspective: External (customer).
Description: How an improvement activity will affect the software product in question with respect to the criteria for evaluation (quality attributes).
Criteria for evaluation: Usability (understandability, learnability), reliability, portability, maintainability.

<table>
<thead>
<tr>
<th>Impact</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>Greatly improves quality of the software product</td>
</tr>
<tr>
<td>+1</td>
<td>Improves quality of the software product.</td>
</tr>
<tr>
<td>0</td>
<td>No affect.</td>
</tr>
<tr>
<td>-1</td>
<td>Degrades quality of the software product.</td>
</tr>
<tr>
<td>-2</td>
<td>Seriously degrades quality of the software product.</td>
</tr>
</tbody>
</table>

For each quality attribute, as listed above, the practitioners were asked to make a judgement regarding the rating on the scale.

B.3. Value Component: Internal Efficiency
Perspective: Company Internal.
Description: Represents the integration of both product’s features and software process deployment for better organizing critical complexities within the production system. Used to estimate the real added value of production processes and final product and its influence on markets.
Criteria for evaluation: Time, cost, quality.
### B.4. Value Component: Delivery Process Value

**Perspective:** Customer Perspective.

**Description:** Quality of process in installing/upgrading/receiving the product with respect to cost, quality and time.

**Criteria for evaluation:** Time, cost, quality.

<table>
<thead>
<tr>
<th>Impact</th>
<th>Time</th>
<th>Cost</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>Greatly reduces delivery time (cut by 26-50%)</td>
<td>Greatly reduces delivery cost (cut by 26-50%)</td>
<td>Greatly improves delivery process quality</td>
</tr>
<tr>
<td>+1</td>
<td>Reduces delivery time (cut by 10-25%)</td>
<td>Reduces delivery cost (cut by 10-25%)</td>
<td>Improves delivery process quality</td>
</tr>
<tr>
<td>0</td>
<td>No affect.</td>
<td>No affect.</td>
<td>No affect.</td>
</tr>
<tr>
<td>-1</td>
<td>Increases delivery time (+10-25%)</td>
<td>Increases delivery cost (+10-25%)</td>
<td>Degrades delivery process quality</td>
</tr>
<tr>
<td>-2</td>
<td>Greatly increases delivery time (+26-50%)</td>
<td>Greatly increases delivery cost (+26-50%)</td>
<td>Seriously degrades delivery process quality</td>
</tr>
</tbody>
</table>

### B.5. Value Component: Human Capital Value

**Perspective:** Company Internal.

**Description:** Human capital refers to the stock of skills and knowledge embodied in the ability to perform labor so as to produce economic value. It is the skills and knowledge gained by a worker through education and experience.

**Criteria for evaluation:** competence, motivation and intellectual agility.
Product management: Product managers are responsible for the product strategy and the road-mapping of product releases and features. For that they have to be in close contact with the market units. In Figure 5 their primary responsibility can be attributed to the high level specification and the prioritization.

System management: System management provides a technical view regarding the system structure, and knows the anatomy well. Hence, they are important in order to determine how new features can be integrated into the existing system anatomy. Hence, their primary responsibility can be allocated to the compound system development box and the anatomy in Figure 5.

Program management: Program management is responsible for the planning and control of system development, hence their responsibility can be connected to the boxes annotated as “System Development” in Figure 5.

Design and coders: Designers and coders implement the solution, hence they work is done within the System Development boxes (see agile team sprint - AT Sprint in Figure 5).

Test (LSV/C-LSV): The LSV team conducts the system test on system level (LSV), and on system of systems level (C-LSV), hence their responsibility is related to the boxes “System Level Test” and “Compound System Test”.

Product packaging: Product packaging uses the outcome of the compound system test and packages the system to be deliverable to the customer (this includes, for example, configuration and installation).
Chapter 6 - Innovative Features Selection using Real Options Theory

Mahvish Khurum, Sebastian Barney
Blekinge Institute of Technology
S-372 25, Ronneby, Sweden
mahvish.khurum@bth.se, sebastian.barney@bth.se
Abstract

Innovation enables product differentiation and supports market growth. However, determining the value of an innovative feature is a difficult task due to the number of risks and uncertainties involved. This paper proposes the use of real options theory to support software product managers to decide whether to make an investment in an innovative feature or not. This approach creates a richer decision space, allowing for more informed decision-making, leading to greater return on potential.

1. Introduction

Requirements selection is a very important phase in the continuous development of a software product [1-4]. Decisions made at this stage have a great impact on the kind of features packaged in a product’s release. In today’s rapidly changing business environment, requirements engineering (RE) has transformed from a traditional bespoke approach towards a market-driven one. Today managers take a more proactive approach and operate strategically, taking early decisions by gathering requirements to improve a product’s market value. Requirements today include innovative features whose potential and value is hard to estimate, in addition to the ones gathered from stakeholders.

Innovations play a major role for any product in the market by creating competitive advantage, opportunities for company to evolve market trends, and expand the client base. But for realizing these innovative features market uncertainty can become a major risk. A strategic and proactive approach in decision making becomes essential and this increases a product manager’s responsibilities. He has to think and act by analyzing various possibilities and perspectives to balance product value, company vision and mission, targeting and creating a market, killing market competition, and enhancing a product’s portfolio; but all in a specified time with available budget and resources [4].

This paper presents how the concept of option space [5] can be used for selecting and planning innovative features given the uncertainty involved and the time left, to take a decision.

The paper is structured as follows. Background to the current method used for determining the value of major investments and real options is presented in Section 2. Section 3 contains an example showing how real options can be used for making decisions about innovative features. Section 4 summarizes the paper.

2. Background

Today the most established method for valuing long-term investments is net present value (NPV) [6, 7]. This metric uses estimated cash flows (in and out), adjusted against the risk-free interest rate to determine if an investment will turn a profit. If NPV is positive the investment is expected to make a profit, while if it is negative it is expected to loose money, and a result of zero translates to breaking even.

There are a number of problems with NPV that make it unsuitable for determining the value of some investments under uncertainty [8]. Perhaps most problematic with regard to
innovation as it assumes risk levels and returns are static, despite this rarely being the case when dealing with innovative features.

NPV also assumes two possible scenarios for a given investment opportunity – invest now in full or never invest [8]. It is important to recognize that many investments do not need to be made immediately – companies should not only decide if an investment should be realized, but when it should be realized and to what extent. It is possible to reduce risk by gathering more information before investing the full amount of required funds through activities like prototyping, research, experiments and market surveys. Each of these activities provide a company with more information with which to make future investment decisions – allowing for more informed choices to be made, whether the company should proceed or cancel an ongoing investment.

Recognizing the problems with NPV, a number of software economics researchers have proposed the application of real options theory to the software development domain as a possible alternative to NPV [7-12].

Real options theory is based on financial options. A financial option is an agreement between two parties. It gives one party the right, but not the obligation, to buy or sell an asset at, or before, a specified future date and at a set price from the other party. The major differences between real and financial options are that the real options are internal to one company and they concern “real” and not financial assets [8].

Options-thinking brings with it a sophisticated valuation method that addresses a number of problems with NPV [8] – the Black-Scholes pricing model. This model provides decision makers with richer information than NPV, as it takes into account:

- The uncertainty of expected cash flows,
- The present value of expected cash flows,
- The value lost over duration of option,
- The risk-free interest rate,
- The present value of fixed costs, and
- The time to expiry of the option.

The Black-Scholes model for pricing options includes all of the information required for the NPV calculation, but also metrics on volatility and the amount of time it is possible to delay making a decision. It is also possible to visualize option space in a two-dimensional graph, as shown in Figure 1 – with expected profitability on the x-axis and the volatility on the y-axis [5].

It is possible to split the graph of option space into a number of regions [5]. When the volatility of an investment is low an investment decision can be made whether the value-to-cost ratio is greater than one (invest now, Region 1 in Figure 1) or less than one (do not invest, Region 6 in Figure 1). The value-to-cost ratio defines whether the option is expected to make a profit by dividing the expected cash inflow by the expected cash outflow, adjusting for the risk-free interest rate. The volatility of an investment will decrease over time as more becomes known about the investment.

Looking at Figure 1, if the value-to-cost ratio is greater than one, it is expected that the investment will be profitable. However, increasing volatility means that there is a higher risk this will not be the case. Thus, we can divide these options into two cases – ones in which to maybe now invest (Region 2 in Figure 1), and one in which to probably later invest (Region 3 in Figure 1). Given that the volatility will decrease as the investment gets closer to the exercise date; it is possible to delay the investment decision to later.
However, investments with a value-to-cost ratio that is less than one should not be instantly discarded. Just as higher volatility means some investments perceived as profitable may not be, it also means that some investments perceived as unprofitable may turn out profitable. The more profitable and volatile investments have the greatest chance of becoming profitable and can be considered for investment *maybe later* (Region 4 in Figure 1).

It is important to manage investment options, even prior to a decision on whether they should be exercised [5]. Over time options will naturally decrease in volatility and value, however, it is possible to nurture these investments to increase their profitability. Working to reduce costs, increase sales, or reduce fixed costs all help increase the profitability of an investment prior to deciding whether or not to make the investment.

While Racheva et al. [13] have discussed application of real options for requirements prioritization in the projects’ context, the approach presented here particularly discusses long term investments in innovative features instead of agile requirements prioritization in projects.

### 3. Real Options for Making Decisions about Innovative Features

Using this understanding of real options, it is possible to define innovative features as a specific type of real options, and exploit the benefits and management opportunities that come with it. This section illustrates the use of real options in a product management scenario.

The definitions of factors used for calculating the value of real options for the innovative features (exemplified in the next section) are given in Table 1.

#### 3.1 Example Scenario

There is a leading telecommunication company that provides 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.

The research and development department has forecasted trends for the coming years as innovative features. The product management wants to decide which innovative features (or options) should be implemented (utilized) or discarded now, and which ones to delay introducing. The features marked as F1, F2…F6 are described as below [14].
F1 - **Sensors everywhere:** The mobile phone has a lot of information about the world around its user. If we take that information and combine it with the information from every other phone, it is possible to provide an snapshot of what is going on in the world right now and offer this to the user.

F2 - **Smart alerts:** The phone is aware of your situation and alerts its user when something needs his/her attention.

F3 - **Tool for economic development:** The phone can be more than just a convenience; it can be the user’s livelihood. For example, in South Africa, sugar farmers can receive text messages advising them of how much to irrigate their crops.

F4 - **The future-proof device:** Just as the Internet already has made it easy to provide users with access to content and services, opening up the phone can make it easy for developers to create and improve applications and content for the mobile market. The applications and services that a user cares about can then get automatically installed on your phone.

F5 - **Safer software through trust and verification:** The phone provides tools and information to empower its user to decide what to download, what to see, and what to share. Trust is the most important currency in the always connected world, and the phone can help a user stay in control of his/her information.

F6 - **Augmented reality:** The phone uses its arsenal of sensors to understand its user’s situation and provide him/her with information that might be useful.

Table 2 shows categorization of F1-F6 in the option space based on value-to-cost metrics, NPV and volatility metrics. Note that each of them involves assets of $5 million. Two of them (F2, F3) require capital expenditures of $4 and $3 respectively; the other four require expenditures of more than the asset value. Consequently, F2 and F3 have positive NPVs of $1 million and $2 million respectively whereas each of the other four has negative NPV. Conventional capital budgeting offers only two prescriptions – invest or do not invest which when applied in this scenario means accept F2 and F3 and reject all the others.

Although their NPVs are very close, the six features (F1, F4, F5, and F6) have different time to expiration and volatility profiles, which gives different values for their value-to-cost and volatility metrics. This makes them fall into different regions of the option space.

### Table 1 – Factors affecting the value of an option

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Underlying asset value (S)</td>
<td>Current value of implementing a feature</td>
</tr>
<tr>
<td>Exercise price (X)</td>
<td>Cost of implementing a feature including the cost of legacy migration</td>
</tr>
<tr>
<td>Volatility (σ)</td>
<td>Uncertainty of customers demand</td>
</tr>
<tr>
<td>Time to expiration (T)</td>
<td>Market window for a feature before it is no longer demanded</td>
</tr>
<tr>
<td>Net Present Value (NPV)</td>
<td>The underlying asset value (S) minus the exercise price (X).</td>
</tr>
<tr>
<td>Value to cost metric (NPV_q)</td>
<td>The value of implementing a feature divided by the present value of the expenditure required.</td>
</tr>
<tr>
<td>Volatility metric (σ(T)^1/2)</td>
<td>Measure of level of changes before an investment decision must be made.</td>
</tr>
</tbody>
</table>
F1 is a never feature in Region 6, F2 falls into Region 1. Since for both of them time has run out making volatility metric 0. From Table 2 it is possible to see that F3 seems to be promising as its NPV is positive and its NPVq is greater than one. This makes it fall into Region 3 meaning that it can be considered for early implementation (exercised). However, unless there is some predictable loss in future value (either a rise in development cost or decrease in value), then early implementation is unnecessary and suboptimal. F4’s NPVq is greater than 0, but its NPV is less than zero. As a result it falls into Region 3 and is very valuable as an option, despite its negative NPV. This is due to the fact that it will not expire for four years and has quite a high volatility. Calculations in Table 2 show that F5 falls into Region 4 with two years to go and the moderate \( \tilde{\sigma} = 0.3 \) per year, it just might make it. F6 seems to be less promising as the decision has to be made in less than a year, the volatility is low, and thus there is not much probability that F6 will become profitable before the time runs out.

As a real options based frameworks can account for flexibility and uncertainty, it produces a different estimation of the features to the conventional NPV methods. It can be evidently seen that NPV method gives a value of $3 million, while option-pricing gives a value of $4.61 million, which is almost double. By placing the innovative features in different regions of Table 2 – Vital Statistics for Feature F1-F6

<table>
<thead>
<tr>
<th>Variables</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Underlying asset value ($ million)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>X</td>
<td>Exercise price ($ million)</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>T</td>
<td>Time to expiration</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>( \tilde{\sigma} )</td>
<td>Standard deviation (per year)</td>
<td>.3</td>
<td>.3</td>
<td>.3</td>
<td>.8</td>
<td>.3</td>
</tr>
<tr>
<td>( R_f )</td>
<td>Risk-free rate of return</td>
<td>.05</td>
<td>.05</td>
<td>.05</td>
<td>.05</td>
<td>.05</td>
</tr>
<tr>
<td>( \text{NPV}_q )</td>
<td>Value to cost metric</td>
<td>.78</td>
<td>1.25</td>
<td>1.83</td>
<td>1.86</td>
<td>.919</td>
</tr>
<tr>
<td>( \tilde{\sigma}(T)^{1/2} )</td>
<td>Volatility metric</td>
<td>0</td>
<td>0</td>
<td>.424</td>
<td>1.6</td>
<td>.424</td>
</tr>
<tr>
<td>C</td>
<td>Call value($ million)</td>
<td>0</td>
<td>1.00</td>
<td>2.023</td>
<td>1.02</td>
<td>.109</td>
</tr>
<tr>
<td>S-X</td>
<td>Conventional NPV ($ millions)</td>
<td>-2</td>
<td>1</td>
<td>2</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>Region in option space</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Exercise decision</td>
<td>Never</td>
<td>Now</td>
<td>Maybe Now</td>
<td>Probably later</td>
<td>Maybe later</td>
<td>Probably never</td>
</tr>
</tbody>
</table>

which features to select/reject for implementation now and wait for certain features as their prospects can differ with time.

4. Summary

The very nature of innovation makes its very uncertain which innovative features to implement. While planning for innovative features, the use of real options provides the following benefits:
1. It provides a valuation method that can be used to analyze different future possibilities and make decisions other than now or never.
2. With the use of a volatility metric, uncertainty is explicitly taken into account, something often missing in current decision making.

3. These valuations can be repeated when new features are proposed and/or when there are changes in market and technology trends that can affect the volatility metric and NPV\textsubscript{q} metric to position/reposition features in the options space. For promising features in Regions 3 and 4 managers can proactively try to increase their value, for example by decreasing costs, saving taxes and/or through marketing campaigns.

4. The use of real options gives a homogenous decision support material that can be understood and used by all stakeholders from revenue management to product management to developers.

The concept of real options is not perfect, and displays some of the shortcomings of the other tools (for example financial options). However, the use of real options shown through an example in this paper seems usable and useful. Future work involves piloting the idea and refining it for the specific domain of software product development.

References

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 the diverse understanding of issues such as priority, consequence of realization, and interpretations of strategy as pertaining to the short-term and long-term development of software intensive products. 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 a company. There is a need to develop the functionality that is strategically most significant, while satisfying customers and being competitive, time efficient, cost effective, and risk minimizing. In order to achieve a balance between these factors, all the stakeholders, within an organization, need to agree on the strategic aspects and value considerations to be considered, and their corresponding relative importance.

Objective: The objective of this thesis is to provide enhanced decision support for product managers faced with decision-making challenges. This involves, but is not limited to, enhancing the alignment between the product and portfolio management with respect to product strategies, and enabling the use of value as a basis for product management and development related decisions.

Method: A number of empirical studies, set in industry, have been performed. The research methods used span from systematic mapping, and systematic reviews to case studies, all aligned to identify possibilities for improvement, devise solutions, and incrementally evaluate said solutions. Close collaboration with industry partners was at the core of the research presented in this thesis.

Result: The MASS method presented in this thesis can be used to evaluate strategic alignment and identify possible root causes for misalignment. To strengthen strategic alignment, the Software Value Map and corresponding decision support material, proposed in the thesis, can be used by product managers for making effective and efficient strategic decisions in relation to portfolios, products and process improvement, following a systematic and aligned process.

Conclusions: 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 identify the main challenges. The decision support developed takes the form of maps and frameworks that support software product management on product and portfolio level decisions, strategic alignment, value-based requirements selection, and value-based process improvement.