“RiPLE-SC: An Agile Scoping Process for Software Product Lines”

Marcela Balbino Santos de Moraes

M.SC. DISSERTATION

RECIFE, AUGUST/2010
Universidade Federal de Pernambuco  
Centro de Informática  

Marcela Balbino Santos de Moraes  

“RiPLE-SC: An Agile Scoping Process for Software Product Lines”  

Trabalho apresentado ao Programa de Pós-Graduação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.  

A M.Sc. Dissertation presented to the Federal University of Pernambuco in partial fulfillment of the requirements for the degree of M.Sc. in Computer Science.  

Advisor: Silvio Romero de Lemos Meira  
Co-advisor: Eduardo Santana de Almeida  

Recife – PE  
August, 2010
Moraes, Marcela Balbino Santos de
*
ix, 164 folhas : il., fig., tab.


Inclui bibliografia e apêndice.

1. Engenharia de software. 2. Reuso de software. 1. Título.

005.1 CDD (22. ed.) MEI2010 – 0151
Dissertação de Mestrado apresentada por Marcela Balbino Santos de Moraes à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “RiPLE-SC: An Agile Scoping Process for Software Product Lines”, orientada pelo Prof. Silvio Romero de Lemos Meira e aprovada pela Banca Examinadora formada pelos professores:

Prof. Silvio Romero de Lemos Meira
Centro de Informática / UFPE
Acknowledgments

There were a lot of people that directly or indirectly participated for the results of this work. Without all them, I could not go the entire Master path. My excuses for the ones I forgot, it does not mean you were not important.

Initially, I would like to thank to all who helped me to start the Master, in special to Liliane dos Santos Machado for recognizing my abilities and inserting me in the scientific research.

Next, I would like to thank the Reuse in Software Engineering (RiSE) Labs for giving me the support and space to discuss the aspects that were useful in my work. In special, my co-advisor, Eduardo Almeida, for his patience in giving me the directions for my Master. Luanna Lopes, Flávio Medeiros, Heberth Braga, Paulo Silveira, Vanilson Buregio, Leandro marques, Ivonei Freitas, Ivan Machado, Jonatas Bastos, Iuri Santos, Leandro Oliveira, Vinicius Garcia and Raphael Pereira for their revisions, thoughts and comments. My advisor, Silvio Meira, for accepting me as his student.

I would like to thank to all who participated in the evaluation of this work, thanks for their dedication and patience. Moreover, I would like to thank MedicWare System for the opportunity to offer the environment and all resources to perform the case study.

There are people that I desire special thanks, friends who helped me in difficult moments such as, Rosane e Daniele. My parents, who supported me in all the decisions, in special my dear mother, Elza, my father, Ricardo and my brother Diego.

I would like to thank to my fiancé for his respect and love.
My gratitude to my family is immeasurable and for them I dedicate this work. However, I dedicate this dissertation, in special, for my cousin, Marielly, a sister who I passed away few months and for my aunt Ednaldo who also passed away.

Finally, I would like to thank God for giving me the wisdom and force to perform this work.
Resumo

As constantes mudanças e a busca por novos benefícios na indústria de software possibilitam o surgimento de novas áreas de pesquisa. Neste contexto, uma tendência que tem apresentado importantes benefícios é a área de desenvolvimento ágil de linhas de produtos de software. Linhas de produtos de software é uma importante estratégia de reuso para minimizar custos e tempo de entrega das aplicações, além de maximizar a qualidade e a produtividade do desenvolvimento de software. Entretanto, desenvolver linhas de produtos requer esforços e custos iniciais para apresentar resultados significativos, uma vez que envolve planejamento sistemático, gerenciamento de pontos comuns e variáveis dos produtos e design flexível e detalhado, aspectos que aumentam sua complexidade. Por outro lado, a abordagem ágil apresenta como foco o código e tem seu desenvolvimento realizado de forma iterativa. Além disso, esta metodologia encoraja práticas e valores como comunicação face a face, pequenas iterações, planejamento e design simples e incremental. Comparando linhas de produtos de software e métodos ágeis, diferentes aspectos são identificados. No entanto, pesquisas recentes mostram similaridades entre elas, como: aumento de qualidade e produtividade no desenvolvimento de software e redução de custos e tempo de entrega das aplicações. Portanto, visando diminuir custos e esforços iniciais necessários para adoção de uma linha de produtos de software, este trabalho propõe um processo de escopo ágil para linhas de produtos de software, unindo os benefícios das duas abordagens. O processo proposto é avaliado através de um estudo de caso industrial.

Palavras chave: Linhas de Produto de Software, Escopo, Métodos ágeis, Processo
Abstract

The constant changes and the search for new benefits in the software industry make possible the emergence of new research areas. In this context, a trend that has presented important benefits is the area of agile development of software product lines. Software product lines is an important reuse strategy to minimize costs and time-to-market, as well as to maximize quality and productivity of the software development. However, developing product lines requires initial efforts and costs to present significant results, since it involves systematic planning, management of variabilities and commonalities of the products and detailed and flexible design, aspects which increase its complexity. On the other hand, the agile approach presents as focus the code and has its development performed in an iterative way. Besides, this methodology encourages practices and values as communication face-to-face, short iterations, simple and incremental planning and design. Comparing software product lines and agile methods, different aspects are identified. However, recent researches show similarities among them, such as: increase of quality and productivity in software development and reduction of costs and time-to-market of the applications. Therefore, aiming to decrease initial costs and efforts necessary for adopting a software product line, this work presents an agile scoping process for software product lines, joining the benefits of the two approaches. The presented process is evaluated through an industrial case study.

Keywords: Software Product Lines, Scoping, Agile Methods, Process
# Table of Contents

Acknowledgments .................................................................................................................. i
Resumo .................................................................................................................................. iii
Abstract ................................................................................................................................. iv
Table of Contents .................................................................................................................. v
List of Figures ........................................................................................................................ viii
List of Tables ........................................................................................................................ ix

## Introduction

1.1. Motivation .......................................................................................................................... 2
1.2. Problem Statement .......................................................................................................... 3
1.3. Overview of the Proposed Solution .................................................................................. 4
  1.3.1. Context .......................................................................................................................... 4
  1.3.2. Proposal Outline ......................................................................................................... 6
1.4. Out of Scope ..................................................................................................................... 6
1.5. Statement of the Contributions ....................................................................................... 7
1.6. Organization of the Dissertation ..................................................................................... 7

## An Overview on Software Product Lines

2.1. Introduction ....................................................................................................................... 10
2.2. Software Product Lines .................................................................................................. 10
  2.2.1. SPL Essential Activities ........................................................................................... 12
  2.2.2. SPL Adoption Strategies ......................................................................................... 14
2.3. Scoping for Software Product Lines ............................................................................... 14
  2.3.1. Scoping Types ........................................................................................................... 15
  2.3.2. Specific Techniques .................................................................................................. 16
  2.3.3. Risks Related to Scoping ......................................................................................... 18
2.4. Chapter Summary .......................................................................................................... 19

## A Systematic Review on Software Product Lines Scoping

A Systematic Review on Software Product Lines Scoping ......................................................... 20
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>RiSE Labs influences</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>RiSE Labs projects</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Economics of software product line engineering (Van der Linden et al., 2007)</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Time-to-market with and without product line engineering (Pohl et al., 2005)</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>SPL essential activities</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>Core asset development (Clements and Northrop, 2001)</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>Product development (Clements and Northrop, 2001)</td>
<td>13</td>
</tr>
<tr>
<td>2.6</td>
<td>Context diagram (Clements, 2005)</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Percentage of the scoping types</td>
<td>36</td>
</tr>
<tr>
<td>4.1</td>
<td>Scrum. Adapted of Hunt (Hunt, 2006)</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Flow of the FDD</td>
<td>57</td>
</tr>
<tr>
<td>5.1</td>
<td>RiPLE</td>
<td>64</td>
</tr>
<tr>
<td>5.2</td>
<td>Relationship of the SC with RM, EM and RE</td>
<td>69</td>
</tr>
<tr>
<td>5.3</td>
<td>Phases of the RiPLE-SC</td>
<td>72</td>
</tr>
<tr>
<td>5.4</td>
<td>Pre-scoping phase</td>
<td>74</td>
</tr>
<tr>
<td>5.5</td>
<td>Domain scoping phase</td>
<td>79</td>
</tr>
<tr>
<td>5.6</td>
<td>Product scoping phase</td>
<td>85</td>
</tr>
<tr>
<td>5.7</td>
<td>User story (Beck and Andres, 2004)</td>
<td>86</td>
</tr>
<tr>
<td>5.8</td>
<td>Product map</td>
<td>91</td>
</tr>
<tr>
<td>5.9</td>
<td>Reuse measurement methods (Almeida, 2009)</td>
<td>93</td>
</tr>
<tr>
<td>5.10</td>
<td>Assets scoping phase</td>
<td>93</td>
</tr>
<tr>
<td>6.1</td>
<td>Correlation among products</td>
<td>105</td>
</tr>
<tr>
<td>6.2</td>
<td>Work flow of the MedicWare</td>
<td>106</td>
</tr>
<tr>
<td>6.3</td>
<td>Timeline of the project phases</td>
<td>117</td>
</tr>
<tr>
<td>6.4</td>
<td>Satisfaction level</td>
<td>119</td>
</tr>
<tr>
<td>6.5</td>
<td>Effort distribution during the project</td>
<td>120</td>
</tr>
<tr>
<td>6.6</td>
<td>Scoping percentage for each phase of the RiPLE-SC in the case study</td>
<td>120</td>
</tr>
</tbody>
</table>
List of Tables

Table 3.1 Summary of the aspects of the scoping approaches ........................................... 43
Table 4.1 Summary of agile methods .................................................................................. 49
Table 4.2 Comparing SPL and AM (Qumer and Henderson-Sellers, 2007; Hanssen and Fægri
2008) .................................................................................................................................. 61
Table 5.1 Summary of the RiPLE-SC ................................................................................ 71
Table 5.2 Quality attributes .................................................................................. 88
Table 5.3 Features list ........................................................................................................ 89
Table 5.4 Benefit and characterization metrics (Almeida, 2007) ........................................ 95
Table 6.1 Stakeholders and roles of the case study ............................................................. 110
Table 6.2 Cost metric ......................................................................................................... 115
Table 6.3 Complexity metric .............................................................................................. 116
Table 6.4 Aggregate value metric ...................................................................................... 116
Table 6.5 Members’ profile ................................................................................................. 118
Introduction

The main aspect for the adoption of a software development process, using objects or not, is the premise of reutilization. The idea is to build software across the utilization of existing artifacts or knowledge (Szyperski, 2002).

The implications for decreasing development time and costs and the increase of the product quality make the reuse approach highly attractive. However, these efforts are often related to individuals and small groups, who practice it in an ad-hoc way, with high risks that can compromise future initiatives in this direction. Nevertheless, currently, organizations are changing this vision, and software reuse starts to appear in their business agendas as a systematic and managerial process, focused on application domains, based on repeatable and controlled processes, and concerned with large-scale reuse, from analysis and specification to code and documentation. In this context, one of the approaches with significant increase of success are software product lines (Weiss et al., 2006) which achieve benefits such as high productivity and high quality, low cost and less time-to-market, maximizing the economical values obtained with software reuse.

Another practice of success on the industry and academy is the agile development. Initial evidences from the results of ongoing research suggests that agile methods are advantageous, given the right context (Boehm, 2002; Erickson et al., 2005; Dyba and Dingsoyr, 2008). Agile development methods encourage strong business involvement in development activities, focus only on the requirements at hand, and deem huge investment in requirement and design up-front unjustifiable.
Software product lines and agile development methods, both approaches have the same overall objective: improve software development efficiency (Hanssen and Fægri, 2008). In this sense, recently, there is an interest in identifying if these approaches can be combined and how the agile aspects can be integrated with software product lines processes for maximizing the benefits searched by software product lines and agile methodologies (Cooper and Franch, 2006; Cooper and Franch, 2008; Ghanam et al., 2009; Ghanam and Maurer, 2010).

In fact, the union between software product lines and agile methodologies can have significant advantages: one is the fact that agile software product lines can be used on a wider variety of projects, such as safety or security critical systems, larger-scale, or with distributed development teams. Furthermore, agile product lines enable to develop a set of similar systems from common core assets, which are like those in software product lines but are simpler, flexible and adaptive (Tian and Cooper, 2006).

However, in spite of agile software product lines present applicability and benefits, specific topics regarding them have not been widely explored, leaving an open gap for the researchers. The scope definition in an agile and incremental way is an example of a field that needs to be investigated and discussed in academy.

Thus, the objective of this dissertation is in providing an agile scoping process for software product lines that aims to integrate the agile aspects with product lines scoping for making possible the maximization of the potential that the union among these two approaches can offer.

This Chapter contextualizes the focus of this work and starts by presenting its motivation and a clearer definition of the problem.

1.1. Motivation

During the last decade, several efforts were conducted to achieve effective ways of dealing with the software industry competitive needs. The competitiveness increases according to industry ability to quickly react to market changes and users requirements (Atkinson et al., 2002). In this context, Agile Product Line Engineering (APLE) is a new research area that aims to join the benefits of product lines and agile methodologies and which was motivated by the need to
rapidly deliver high quality software that meets the changing needs of stakeholders. This vision started to be idealized in 2006 with the publication of some work (Carbon et al., 2006; Tian and Cooper, 2006) which analyzed the viability of integration between product lines and agile methodologies. Moreover, in 2006 was also performed the first International Workshop on Agile Product Line Engineering (APLE).

In the agile product lines scoping field, only one work discuss how to perform agile scoping (Noor et al., 2007). This work introduces collaborative practices from the area of collaboration engineering (Briggs et al., 2006) for obtaining benefits of agile methods such as: constant feedbacks, stakeholders’ involvement and value-based prioritization. However, the approach addressed does not present a complete and systematic way, with sufficient guidance to cover the whole scoping lifecycle. Besides, the approach is not clear regarding the iterative, incremental and evolutionary aspects, essentials characteristics in agile methods. In this context, an important motivation for this dissertation is the need of a process with a well-defined, complete and systematic set of activities incorporating the agile aspects in order to maximize the benefits that can be obtained with the integration of product lines and agile methods.

1.2. Problem Statement

A process can be understood as a collection of related tasks leading to a product (Ezran et al., 2002), or simply, a way of defining specifically who does what, when, and how (Fayad, 1997). A process is important to define how an organization is supposed to perform its activities, and how people work and interact, in order to ensure efficiency, reproducibility, and homogeneity (Ezran et al., 2002). Several work discuss the benefit and improvements of using a well-defined software process (Basili et al., 1995), (McConnell, 1998).

In the context of software product lines, a scoping process with tasks, roles, inputs, outputs of each step in a systematic way is essential, because software product lines need of a systematic planning to obtain significant results. In addition, combining these characteristics with agile aspects makes possible to decrease the up-front costs as well as time-to-market and, consequently, increase the economical potential, because the insertion of agile practices such as communication face-to-face, collaborative work between
technical persons and business persons, workshops, into software product lines scoping, favors the discussion of information high amount of different view point in few time space, as well as decreases the documentation need and increase the simplicity of the planning, giving to the process a lightweight characteristic.

Thus, motivated by the scenario presented in the previous section, the goal of the work described in this dissertation can be stated as:

This work defines an agile scoping process providing phases, tasks, inputs, outputs, roles and guidelines integrated to agile values and principles for a simple and agile scope definition for the software product lines context in an efficient way.

1.3. Overview of the Proposed Solution

In order to accomplish the goal of this dissertation, RiPLE-SC (RiSE Product Line Engineering Process - Scoping) is proposed. This Section presents the context where it is regarded and outlines the proposed solution.

1.3.1. Context

This dissertation is part of the Reuse in Software Engineering (RiSE) Labs¹ (Almeida et al., 2004), formerly called RiSE Project, whose goal is to develop a robust framework for software reuse in order to enable the adoption of a reuse program. RiSE Labs is influenced by several areas, such as software measurement, architecture, quality, environments and tools, and so on, in order to achieve its goal. The influence areas are depicted in Figure 1.1.

![Figure 1.1 RiSE Labs influences](http://www.rise.com.br/research)

¹ http://www.rise.com.br/research
Based on these areas, the RiSE Labs is currently divided in several projects related to software reuse, as shown in Figure 1.2:

- **RiSE Framework**: It involves reuse processes (Almeida et al., 2005), component certification (Alvaro et al., 2006) and reuse adoption and adaptation processes (Garcia et al., 2008).

- **RiSE Tools**: Research focused on software reuse tools, such as the Admire Environment (Mascena et al., 2006), the Basic Asset Retrieval Tool (B.A.R.T) (Santos et al., 2006), which was enhanced with folksonomy mechanisms (Vanderlei et al., 2007), semantic layer (Durao, 2008), facets (Mendes, 2008) and data mining (Martins et al., 2008), the Legacy InFormation retrieval Tool (LIFT) (Brito et al., 2008), the Reuse Repository System (CORE) (Burégio et al., 2008), and the Tool for Domain Analysis (ToolDAy) (Lisboa, 2008).

- **RiPLE**: Stands for RiSE Product Lines Engineering Process and aims at developing a methodology for Software Product Lines, composed of scoping, requirements engineering (Neiva, 2009), risks management (Lobato, 2011), design, implementation, test (Silveira, 2010), and evolution management (Burgos, 2009).

- **SOPLE**: Development of a methodology for Software Product Lines based on services (Medeiros, 2010) and (Ribeiro, 2010), with some idea of RiPLE.

- **MATRIX**: Investigates the area of measurement in reuse and its impact on quality and productivity.

- **BTT**: Research focused on tools for detection of duplicated change requests (Cavalcanti, 2009).

- **Exploratory Research**: Investigates new research directions in software engineering and its impact on reuse.

- **CX-Ray**: Focused on understanding with empirical data the C.E.S.A.R², its processes and practices in software development, including reuse.

² C.E.S.A.R - Recife Center for Advanced Studies and Systems - http://www.cesar.org.br
This dissertation is part of the RiPLE project and its goal is to support the agile scoping in the software product lines context.

1.3.2. Proposal Outline
The goal of this dissertation is to define agile scoping in the context of software product lines, by defining a systematic process integrated to the agile values and principles. The process addresses four main phases: Pre-Scoping, Domain Scoping, Product Scoping and Assets Scoping, allied to the main aspects defined by agile methods such as communication face-to-face, customer has critical role, collaborative participation, simplicity, constant feedback, workshops, and timeboxes (time interval which cannot be exceeded).

1.4. Out of Scope
As the proposed process is part of a broader context (RiPLE), a set of related aspects will be left out of its scope. Thus, the following issues are not directly addressed by this work:

- **Evolution Management.** The software product lines evolution control is ensured by appropriate practices of change management, including changes in scoping artifacts. These practices are part of the Evolution Management process, thus they will not be considered in this process. They are part of another process in RiPLE, the RiPLE-EM (Burgos, 2009).
• **Risk Management.** Risk management is important in all phases of software product lines. The practices and techniques for this management are part of the Risk Management process of the RiPLE that is in phase of definition as a PHD thesis.

• **Tool Development.** As usual in software engineering, also in scoping, the question on tooling comes up. However, the construction of a collaborative tool for scoping can be complex and it is out of the scope of this work.

### 1.5. Statement of the Contributions

As a result of the work presented in this dissertation, the following contributions can be enumerated:

• A systematic review on scoping for software product lines (Moraes et al., 2009), in an attempt to understand this research area and identify trends to follow.

• The definition of an agile scoping process for software product lines.

• The definition, planning and analysis of an industrial case study in order to evaluate the proposed process.

### 1.6. Organization of the Dissertation

The remainder of this dissertation is organized as follows:

• Chapter 2 discusses software product lines basic concepts and activities, as well as the scoping field. The relation between software product lines and scoping is also discussed.

• Chapter 3 presents a systematic review on existing approaches for software product lines scoping with the goal of mapping them to better understand the state-of-the-art in this field and serve as basis for the proposed process.

• Chapter 4 presents an overview on the main agile methods with the objective of analyzing the benefits, challenges and existing values, principles and techniques. It also makes a parallel between agile methods and product lines.

• Chapter 5 describes the proposed process to define agile scoping for product lines, with the existing phases, tasks, roles, inputs and outputs.
• Chapter 6 presents an industrial case study which evaluates the use of the proposed process.
• Chapter 7 concludes this dissertation by summarizing the findings and proposing future enhancements to the propose, along with the concluding remarks.
• Appendix A presents the result of the quality score for each work analyzed in the systematic review.
• Appendix B presents the template for documentation of the work analyzed in the systematic review.
• Appendix C describes the templates related to the artifacts defined in the process.
• Appendix D presents the questionnaires used in the evaluation of the process.
• Appendix E presents the meeting minutes correspondents at the meetings performed during the evaluation of the process.
In the highly competitive world, companies face a major threat: they must meet a demand for more and more specialized and customized products (Sanderson and Uzumeri, 1997). The mass customization was initially introduced in the domain of automobiles by Henry Ford and enabled production for a mass market cheaper than individual product creation on a handcrafted basis. The same idea was made also by Boeing, Dell, and even McDonald’s (Northrop, 2002).

Initially, the customers were satisfied with standardized mass products (Pohl et al., 2005), however, not all people want the same kind of car for any purpose. Thus, industry was confronted with the rising interest for individualized products, which was the beginning of mass customization. However, for the companies, the mass customization meant investment in new technologies and increase of production costs. In this way, many companies started to introduce common platforms for their different types of products, by planning beforehand, which parts would be used in different product types. This enabled the decrease of costs for the companies and is the key for software product lines.

Product lines are characterized as a systematic way for achieving reuse. In this context, scoping is the key for planning of what should be reused. Thus, product lines has proven to be the methodology for developing a diversity of software products and software-intensive systems of planned form and with lower costs, in shorter time, and higher quality (Pohl et al., 2005).
In this chapter the main concepts of software product lines will be discussed, emphasizing the scoping phase.

2.1. Introduction
The concept of product lines was fully introduced in the context of software in the early 1990’s. One of the first contributions was the description of the Feature-Oriented Domain Analysis (FODA) method. Approximately at the same time, several companies started to address the issue more systematically. For example, Philips introduced the Building-Block method in the early 1990’s. These first approaches were leveraged by massive investments in Europe in the area of software product line engineering, both inside companies and part of several scientific projects. In this context, product lines succeed because the commonalities shared by the software products can be exploited to achieve economies of production (Clements and Northrop, 2001).

The identification of commonalities and variabilities is the key prerequisite for software product lines. Commonality reveals common characteristics of products in a product line. Variability is a tangible difference among products of a product line that can be revealed and distributed in any phase of development cycle of the product line. Thus, the variabilities are modeled to enable the development of customized products by reusing predefined and adjustable assets (Pohl et al., 2005).

2.2. Software Product Lines
A Software Product Line (SPL) is defined as “a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way” (Clements and Northrop, 2001).

Substantial production economies can be achieved when the systems in a SPL are developed from a common set of assets in a prescribed way, in contrast to be developed separately, from scratch, or in an arbitrary fashion. Thus, the main idea of product lines is that it takes a family perspective instead of a single product perspective, this fact enables large scale reuse across the family
members (Schmid, 2002a). In Figure 2.1 is possible to see the economic benefit that can be obtained with the adoption of product lines.

![Figure 2.1 Economics of software product line engineering (Van der Linden et al., 2007)](image)

Product lines have also high influence on the quality of the resulting software (Van der Linden et al., 2007). As a new product is resulted from a set of tested and matured assets, the defects density of these products can be expected to be drastically lower and consequently the quality level can be considerably increased.

Another very critical success factor for a product line is the time-to-market (Pohl et al., 2005). For product line engineering, the time-to-market indeed is initially higher, because the reuse infrastructure has to be built first. However, after having passed this hurdle, the time-to-market is considerably shortened as many artifacts can be reused for each new product as can be seen in Figure 2.2.

![Figure 2.2 Time-to-market with and without product line engineering (Pohl et al., 2005)](image)
All these benefits previously cited are aspects which motive the adoption of SPL. Furthermore, in the context of product lines, the customers get products adapted to their needs, while that in the past, for example, these customers used a different user interface and a different installation procedure with each new product. The systematic combination of mass customization and common platforms is the key for product lines.

2.2.1. SPL Essential Activities
According to (Clements and Northrop, 2001) SPL is composed of three essential activities intrinsically related to each other in a way that changing one of them implicates in analyzing the impact on the others as shown in Figure 2.3. Each rotating circle represents one of the essential activities, which are linked together and in perpetual motion, showing that all three are essential, are inextricably linked, can occur in any order, and are highly iterative, as shown in Figure 2.3.

![Figure 2.3 SPL essential activities](image)

The activities illustrated in Figure 2.3 are detailed as follows:

- **Core Asset Development:** The goal of this activity is to establish the reusable platform and thus for defining the commonality and the variability to fit different environments and products in the same domain of the product line. Core assets are the basis of product line (Clements and Northrop, 2001). It consists of all types of software artifacts (requirements, design,
realization, tests, so on.). Figure 2.4 shows the core asset development activity in details.

Figure 2.4 Core asset development (Clements and Northrop, 2001)

- **Product Development:** The goal of the product development activity is to derive specific applications by exploiting the variability of the product line and ensure the correct binding of the variability according to the applications’ specific needs. The relationship of the SPL essential activities, with the focus on product development, is illustrated in Figure 2.5.

Figure 2.5 Product development (Clements and Northrop, 2001)

- **Management:**Includes technical and organizational management, where technical management is responsible for the coordination between core asset and product development and the organizational management is responsible for the production constraints and production strategies.
2.2.2. SPL Adoption Strategies

There are three widely recognized approaches that a software product line organization can pursue with respect to investments in core assets. These depend on the context in which the product line organization operates (McGregor, 2008):

- **Proactive:** In the proactive approach, the reusable assets are developed prior to any product development. A product line requirements model and its architecture are created and from these assets code are derived. This approach is efficient in a very stable domain where the organization has lots of experience. One of the biggest risks involved with this approach includes changes in business rules or in domain/product definitions before any products in SPL had been built. This would make some of the assets already created useless for future products.

- **Reactive:** In the reactive approach, the reusable assets are harvested from products after they are built and deployed. Initially, the product is built like any single-product development effort. As other products are built they use assets harvested from the products that have been built so far. The set of reusable assets evolves into a more useful collection over time. This approach reduces the risk that assets will become obsolete. Every asset is used at least once. The risk is that some short fuse business opportunities will be missed because product production is not as fast as it could be. There is also the risk that lack of a product line architecture will result in lots of reworking of assets to make them suitable for future products.

- **Incremental:** The incremental approach is a compromise among the two extremes. The set of assets is built in scheduled increments. The increments are usually defined to provide assets needed for a set of products scheduled to be produced in the near future. The risks of the other two approaches are present here in reduced form but still present.

2.3. Scoping for Software Product Lines

Product line scoping is a key, if not the key activity in the development and evolution of Software Product Lines (Dalgarno, 2008). It is interesting from a product line point of view because it is much more explicit in that context,
whereas in single-system development the scoping activity is so implicit as to be overlooked. Hence, scoping is often regarded as one of the few activities unique to product lines (Kruchten, 1998).

Scoping is an activity which is directly associated to the success of product lines. It identifies the commonalities that members of the product line share and the ways in which they vary. Identifying commonalities and variabilities is the essence of the product line concept and the essence of scoping. The scoping is essential for determining whether a planned system can be built within the product line and from product line core assets (Clements, 2005).

The goal of scoping is to draw the boundary between in and out in such a way that the product line is profitable. For a product line to be successful, its scope must be defined carefully. Thus, successful product line organizations know that their product lines must be developed according to a well-defined scope.

According to Schmid (2002a), scoping can be defined on an abstract level as a management activity that determines in which lifecycle certain functionality will be developed. Moreover, it defines the range of applications that will be developed within the product line.

2.3.1. Scoping Types
The scoping processes of product lines address scoping according to the goals which they desire to achieve. These goals can be product identification, identification of business goals, assessment or ranking of products and features, product portfolio optimization (product and feature planning plus optimization of the products concerning the business goals), asset planning (identification and planning of asset development), assets optimization (it is related to the components economic evaluation) domain assessment (prioritization of domains for reusable development) and release planning.

The goals of a scoping process are directly associated to the scoping types covered by the process. Schmid (2002) highlights that each one of them is a form of reuse planning, however, each one is applicable at a different stage of the overall lifecycle. According to (Schmid, 2000; Bosch et al., 2001), it is
needed to distinguish three different forms of product lines scoping. These three types of scoping are:

- **Product Portfolio Scoping:** This aims at identifying the particular products that ought to be developed as well as the features they should provide. While it has an impact on the actual reuse opportunities, it is usually driven from marketing aspects. Thus, it is more appropriate as a topic of marketing science than as a software engineering topic.

- **Domain Scoping:** This is the task of bounding the domains (i.e., coherent clusters of functionality) that are supposed to be relevant to the product line. It involves analysis of risks and benefits of the domains as well as reuse potential analysis.

- **Asset Scoping:** This aims at identifying the particular (implementation) components (i.e., functional parts of the product line) that should be developed in a reusable manner.

### 2.3.2. Specific Techniques

In strategic product line development, scoping is indispensable and recognized as an important part of product line technology. In this sense, the following practices are extremely relevant (Clements, 2001; Clements, 2005):

- **Applying the What to Build Pattern:** This pattern is an effective way to establish and understand the scope of a product line. The pattern situates the scoping practice area in the broader context of its interactions with other closely related practice areas: understanding relevant domains, building a business case, market analysis, and technology forecasting. The pattern is applied by iterating through its practice areas until the desired understanding of the scope is achieved. An analyst can apply the pattern quickly to a candidate list of product lines to select the one(s) with the most promise or apply it more deliberately to a selected candidate to probe the soundness of the choice.

- **Examining Existing Products:** Conducting a thorough study of existing products helps form the notions of commonality across a potential product line and identifies the types of differences that are likely to occur. A survey of each group that is developing these products
will likely identify future plans, market strategies, and context. In many cases, existing products will contain potential product line assets that can be mined and used in the future.

- **Conducting a Workshop to Understand Product Line Goals and Products:** It is important to gather the potential product line stakeholders to set the direction for the product. The stakeholders include managers, market expert, developers, users, testers, tool developers, technology researchers, and domain experts. The market analysis and business case will also explore goals; however, during scoping, the activity examines the product line more from the perspective of the user than of the organization. The workshop should also establish a coarse-grained schedule that aligns product line development with marketing or overall mission strategies.

- **Context Diagramming:** A context diagram places the product line in the context of other systems and of product users. This diagram depicts the important entities that affect the product line or are affected by the product line such as: people, physical environment, and other systems. Figure 2.6 shows an example of context diagram in a sound system.

![Figure 2.6 Context diagram (Clements, 2005)](image-url)
• **Developing an Attribute/Product Matrix:** An attribute/product matrix is a relation between the products and the characteristics related to them. Typically, the attributes that drive the market are listed vertically across the left side of the matrix, and the different products are listed horizontally across the top of the matrix. The matrix is used in scoping to define the variability of the product line. By sorting in order of attribute priority, the cluster of the most important attributes that are common across the product can be identified readily.

• **Developing Product Line Scenarios:** According to Clements (Clements, 2001), product line scenarios are the key to defining a scope. They describe user or system interaction with products in the product line. They identify interactions that are common to all products in the product line, as well as those that are unique to a subset of products in the product line.

2.3.3. **Risks Related to Scoping**

Regarding to the risks associated with scoping for SPL, the mains risks are: (Clements and Northrop, 2001):

• **Scope too Small:** if the scope is defined too narrowly, the core assets might not be built in a generic enough fashion to accommodate future growth, thus, new market opportunities can be rejected as being out of scope;

• **Scope too Big:** In cases where the product lines include products members that vary too widely, the core assets will be strained beyond their ability to accommodate the variability, thus, the construction of the product line will have high development effort and the economies of production will be lost;

• **Scope Includes the Wrong Products:** The choice of product and features that maximize the economical potential of the product lines and meet the users concerns is the essential aspect of SPL, thus, the scoping effort must define a scope that reflects both what is desired for the product line from a marketing perspective and what is feasible from a design perspective;
• **Essential Stakeholders do not participate:** The specific practices in scoping require participation from a wide range of stakeholders. The participation of essential stakeholders is required to reduce the obvious risk of defining a scope inappropriate for the product line. Besides, the involvement of a wide range of stakeholders increases the awareness of product line efforts, obtains stakeholders' critical input, and builds momentum for the long-term investment in core asset development and use.

### 2.4. Chapter Summary

Across the years, several studies, including enterprise reports (Bauer, 1993; Endres, 1993; Joos, 1994; Griss et. al, 1995), informal research (Frakes and Isoda, 1994) and empirical studies (Rine, 1997; Morisio et. al, 2002; Rothenberger et. al, 2003) have shown that an efficient form to obtain the reuse benefits is through the application of an efficient reuse process. In this sense, product lines appears as a systematic way for achieving reuse in large scale and has proven its applicability in a broad range of situations, where several benefits are obtained, such as: increase in productivity and product quality, significant reduction of time-to-market and costs and continuous improvement in customer satisfaction through mass customization.

Scoping of product lines is the initial step of SPL and is directly related to the economical success of them. Through scoping, it is possible to identify aspects such as: products that will constitute the product line, risks, reuse potential and costs for implementation of core assets.

Next chapter presents a systematic review on scoping for SPL, discussing current approaches, their weaknesses and strengths, in order to define a base to the proposal of this work.
Software product lines scoping was firstly and largely disseminated through the PuLSE-Eco approach, initially defined by Debaud and Schmid in 1999 (Debaud and Schmid, 1999).

Scoping aims to maximize the economical values obtained from the adoption of product lines, where is possible to identify the feasibility of product lines, since it is responsible for analyzing the potential domains of a given SPL.

This chapter presents eleven SPL scoping approaches representing the state-of-the-art of the area and discussing the requirements and important issues that scoping approaches should consider.

3.1. Introduction
Software Product Lines Scoping (SPLS) is a phase by which information used in software systems development within a domain is identified, captured and organized with the purpose of making it reusable when building new products (America et al., 2001). According to Kruchten (1998), the scoping captures the context, the most important requirements and constraints in order to derive acceptance criteria for the end product. Hence, SPL needs of a very well-defined scoping approach for reducing risks and maximizing the economical potential that can be obtained with them.

Organizations that desire to adopt SPL and need to define scope should identify the most appropriate approach to their context, considering the main activities defined in the approach. Thus, the analysis and comparison among
existing scoping approaches is crucial for selection of an appropriate approach for a specific context.

In this context, this chapter presents a systematic review to investigate the existing approaches on SPL scoping aiming to identify, compare and summarize evidence about the scope definition techniques, analyzing their activities, roles, guidelines, concepts, strong points and drawbacks, and the main features. This study is based on a set of research questions. Moreover, since it is systematically performed and documented, their results are believable to serve as a guide to aid the research community regarding future directions. On the other hand, for practitioners, it can identify the more appropriate approach to be used in industrial scenarios.

This review is systematically performed following Kitchenham's guidelines (Kitchenham and Charters, 2007), which aids in assuring consistent results from individual studies (called primary studies).

### 3.2. Systematic Review Process

A systematic review is a process to identify, evaluate and interpret all available research relevant to a particular research question, or phenomenon of interest (Kitchenham and Charters, 2007). Systematic Reviews are forms of secondary studies, since the purpose is to review all primary studies related to a specific research question, aiming to integrate and synthesize evidences to a specific topic (Kitchenham and Charters, 2007).

Some of the reasons for performing a systematic review are (Kitchenham and Charters, 2007):

1. Review existing evidences about a treatment or a technology;
2. Identify gaps in current research;
3. Suggest issues for further investigation and examine empirical evidence to support or contradicts hypotheses;
4. Provide a framework/background for new research activities; and
5. Know how much confidence the users of systematic reviews can place in the conclusions and recommendations arising from such reviews (Dyba and Dingsoyr, 2008).

A systematic review synthesizes existing work in a manner that is fair, and seen to be fair, and it involves several discrete activities. As stated in
(Kitchenham and Charters, 2007), this study was divided in the following phases:

1. **Planning**: where the need for a review and the protocol for conducting the review are defined.

2. **Conducting**: in which the research is identified, the primary studies are selected and their data extraction and synthesis is performed.

3. **Reporting**: responsible for relating the review steps to the community interested in the study.

Each one of these stages involves iteration, feedback, and refinement defined in the process of the systematic review (Kitchenham and Charters, 2007).

### 3.2.1. Planning

This section presents the stages associated with the initial phase of this systematic review, which includes the protocol definition and the research questions specification and structure.

#### 3.2.1.1. Protocol Definition

The protocol document defines research objectives and how the review will be conducted, which includes the research question definition and the planning on how the sources and studies selection will be carried out. The research questions guide the design of the review process.

Throughout the execution of this systematic review, the protocol was revisited in order to update the data in it contained, since some information was not identified in its first version, but rather along the execution process. The protocol of this review was performed by author of this dissertation and incremental reviews were performed by the RiSE Labs\(^1\) members to be in accordance to the systematic review.

#### 3.2.1.2. Research Questions

As the objective of this study was to understand, characterize and summarize evidence, identifying activities, practical and research issues concerning to research directions in SPL Scoping, we focused our main question on identifying
how the existing approaches deal with scoping in SPL. This was divided in 8 research questions, in order to have a detailed investigation to identify best practices arising from the approaches and which are the essential points to be taken in consideration when defining a SPL Scoping approach.

The research questions are considered the key aspects of the systematic review. In this review, the questions were identified with the investigation of studies in SPL scoping (Schmid, 2002; Clements 2005; John et al., 2006) and with discussion among the RiSE Labs\(^1\) members. In this study, the following research questions were defined:

**Q1. What activities are addressed for the scoping phase in the software product lines approaches?**

The objective of this question is to identify the scoping activities used by the approaches, as well as analyzing the inputs and outputs addressed by each activity.

**Q2. Do the SPL approaches optimize scope?**

This question aims to identify if the approaches optimize scope and which techniques are used for this optimization. Optimizing scope is to adapt a scope to maximize specific objectives (Schmid, 2002), i.e., to align the scope according to the organization’s business goals, such as time-to-market and development effort.

**Q3. What are the types of scope addressed by the approaches?**

The purpose of this question is to identify the types of scope covered by each approach. This question is important to identify the focus of each approach.

**Q4. Which stakeholders are involved in the scope definition approaches?**

This question aims to identify the essentials stakeholders at the scope definition approach and its importance (roles and attributions). According to Clements (2005), the involvement of a wide range of stakeholders increases the awareness of the product line efforts and decreases the risk of defining an inappropriate scope.

**Q5. Do the approaches use specific metrics or cost models for scope definition?**
The goal of this question is to identify if the scope definition approaches use specific metrics, general software metrics or cost models. Moreover, this question aims to identify which types of metrics are used and their importance for the economical value evaluation of the software product line assets.

**Q6. Are the approaches customizable?**

The goal of this question is to identify if the approaches are customizable for different contexts. Besides, the question aims to determine which customization factors are used.

**Q7. Do the approaches treat the new perspective of agile scoping for SPL?**

The goal of this question is to identify if the approaches apply agile aspects in scoping and which techniques, principles or practices are used.

**Q8. How are the approaches related to SPL development?**

The goal of this question is to identify if the approaches have specific activities to define scope according to different development stages, considering situations as: product line development started from the scratch, product line development with existing products and evolution of an existing product line.

### 3.2.1.3. Question Structure

Some criteria should be applied to frame research questions. In this section, these criteria are defined.

According to (Kitchenham and Charters, 2007), three viewpoints are considered for question structure:

1. **Population**, i.e. the people affected by the intervention;
2. **Intervention**, which are usually a comparison among two or more alternative treatments; and
3. **Outcomes**, i.e. the clinical and economic factors that will be used to compare the interventions.

Therefore, these viewpoints were structured in this review of the following form:

1. **Population.** Once this study addresses SPL Scoping techniques, the population is composed of viewpoints from Software Engineers and SPL Professionals working with Scoping;
2. **Intervention.** This review focused on analyzing SPL Scoping approaches; and

3. **Outcomes.** This work intended to identify a set of good practices to build a reliable and effective SPL Scoping process, covering the gaps of the existing approaches and the main challenges for research and development in academia and industry (See Section 3.2.2.3).

### 3.2.2. Conducting

This section discusses the strategy used in the searches, its results, process for studies selection, data analysis and data extraction.

#### 3.2.2.1. Search Strategy

From the research questions some keywords were extracted and used to search the primary study sources: scoping, scope, domain scope, product scope, assets scope, features scope, scope definition techniques, scope definition and scope metrics. All terms were combined with the term “Software Product Line” “OR” “Software Product Family” by using Boolean “AND” operator, to match the search goals. They all were joined each other by using “OR” operator so that it could improve the reliability of the results. Thus this review defines the following search string:

- (scoping OR scope OR domain scope OR product scope OR assets scope OR features scope OR scope definition techniques OR scope definition OR scope metrics) AND (software product family OR software product line)

#### 3.2.2.2. Data Sources

The main data sources of the review were conference proceedings and journals. Books, master and PhD thesis and technical reports were also collected. It is worthwhile to mention that it was not stated any year to limit the search results since our intention was to have a broader coverage of this research field. Only studies written in English were considered.

The initial step was to list important journals published by IEEE, ACM and Springer Link since they can be considered world’s leading interactive

Next, conference proceedings were visited. When available at digital libraries (e.g. ACM and IEEE portals), the web search engine was used in the same way as it was done with journals, however, in some cases it was not feasible to use web search, then the actual proceedings had to be consulted. The searched conferences were: *Annual ACM Symposium on Applied Computing (SAC), Asia Pacific Software Engineering Conference (APSEC), Empirical Software Engineering and Measurement (ESEM), Euromicro Conference on Software Engineering and Advanced Applications (SEAA), European Conference on Software Maintenance and Reengineering (CSMR), European Software Engineering Conference (ESEC), Fundamental Approaches to Software Engineering (FASE), International Computer Software and Applications Conference (COMPSAC), International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), International Conference on Advanced Information Systems Engineering (CAiSE), International Conference on Automated Software Engineering (ASE), International Conference on Composition-Based Software Systems (ICCBSS), International Conference on Information Reuse and Integration (IRI), International Conference on Model Driven Engineering Languages and Systems (MODELS), International Conference on Software Engineering (ICSE), International Conference on Software Engineering and Knowledge Engineering (SEKE), International Conference on Software Reuse (ICSR), International Symposium on Component-based Software Engineering (CBSE), International Symposium on Software Reliability Engineering (ISSRE), Software Product Line Conference (SPLC).*

### 3.2.2.3. Inclusion and Exclusion Criteria

Criteria are defined to establish the reasons for each study to be included or excluded from the review.

In this review, the following inclusion criteria were established:
• SPL approaches which address scope definition. Investigate approaches that show clearly, with specific activities and guidelines, how the scope is identified. The purpose is to investigate which activities are used and their characteristics.

• Studies that present metrics or cost models for scope definition in SPL. This criteria aims to identify how the approaches measure the effort associated to the implementation of a SPL and the costs related to it. By the way, we understand that not all the defined criteria would be presented for the studies and if all criteria were mandatory, the number of selected studies would decrease significantly. Therefore, studies with at least one of the issues addressed in the inclusion criteria were selected for analysis.

The exclusion criteria were:

• Studies with insufficient information about scope definition. Studies that do not have useful information on their activities and techniques for scope definition, besides studies involving only general concepts.

• Duplicate studies. When a study has been addressed by different publications, the most complete version will be used. Studies which do not match the exclusion criteria will be discarded.

3.2.2.4. Quality Criteria

The purpose of the quality criteria is to evaluate the primary studies, as a means of weighting the importance of individual studies guiding their understanding. It is important to notice that the quality criteria refer to the study itself not to the approach presented, meaning that a very well consolidated and experimented approach may not provide well structured information about all the questions asked by the systematic review.

Two types of primary studies can be found in searches: approaches and surveys (reviews) related to the research area. Thus, specific and common quality criteria for both studies types were defined.

The quality criteria defined for analysis of the approaches were: 1. are the
metrics defined? 2. are there guidelines or activities described? 3. are the roles defined? 4. have been the approaches experimented? 5. are the stakeholders defined? 6. is the scope optimized? 7. how well is the study detailed?

The quality criteria defined for analysis of the surveys were: 1. does it define comparison criteria? 2. how clear and coherent is the study? 3. does it define inclusion and exclusion criteria? 4. are all the study questions answered? 5. how adequately has the research approach been documented? 6. how significant are the results?

The score obtained by the approaches in each quality criteria can be: “1”, when the approach satisfies the criteria; “0.5”, when the approach satisfies the criteria partly; and “0”, when the approach does not satisfy the criteria. It is important emphasize that as the approaches address scoping from different perspectives, the quality score does not determinate which approach is best, but identifies which characteristics it satisfies best.

The result of the quality score for each approach is presented in Appendix A.

3.2.2.5. Studies Selection
As soon as the relevant primary studies are obtained, according to the match of the search items and research questions, they need to be assessed for their actual relevance.

The selection process was initiated with the relevant studies identification using defined search items. The found papers were included or excluded based on the inclusion and exclusion criteria.

In the first stage for each selected primary study was applied a brief analysis of the following elements: title, abstract, keywords, conclusion and references. The result of this stage was 27 studies raised from the web search, using the search strategy presented in Section 3.2.2.1 (Search Strategy) applied to the data sources, as shown in Section 3.2.2.2 (Data Sources).

In the second stage a complete analysis of the 27 studies found in the first stage was performed. After this complete analysis and application of our inclusion, exclusion and quality criteria, eleven approaches and two surveys were included in our review. The approaches identified in this review are detailed in the next section.
3.2.2.5.1. Approaches Information

In this section, we present a brief description about the selected studies in a chronological order.

**Feature Scoping for Product Lines (Riebisch et al., 2001)**

In this approach, the features are analyzed according to four levels of priority: **I.** to be implemented for all systems in the product line; **II.** to be implemented for some systems in the product line; **III.** to be implemented for some systems in the SPL in a later development cycle; and **IV.** not to be implemented.

Based on this analysis, calculations of priorities are achieved according to methodologies similar to Quality Function Development (QFD) (Karlsson, 1997), aiming the prioritization of the features. For results interpretation associated with the prioritization, decision tables are used. The decision tables are organized with input values in columns and resulting actions in rows, with conditions in the body of the table. In one step, all actions are performed, where the conditions are met by the input values. If in this step there is no matching action, an optional default action is performed. Using the decision table, a decision about the implementation of each feature of the product line is made.

Through the prioritization and interpretation of the features, this approach enables the management of scoping. However, it is restricted to the definition and prioritization of the features, not identifying domains, products or core assets.

In this study, different situations of SPL development are also considered, these are:

- The start of a SPL development from scratch;
- The start of a SPL development from existing assets; and
- The evolution of an existing SPL.

**Method for Product Line Scoping Based on a Decision-Making Framework (Kishi et al, 2002)**

This study considers scoping as an activity of decision-making. It describes SPL scoping as a set of product lines and product lines as a pair of the members list and architecture.
The appropriateness of sharing the same architecture among multiple products should be examined from two points of view. One is from the point of view of the individual optimal, i.e. whether or not it is good for each product to use the shared architecture, and the other is from that of the whole optimal, i.e. whether or not it is good for the product line as a whole to share the architecture.

The technique shown at this study is well defined and illustrates each activity of the approach in details, besides, it defines a case study with the application of the approach. But, it identifies the architecture before of the scope definition, limiting commonality and variability by architecture.

**A Comprehensive Product Line Scoping Approach and its Validation (Schmid, 2002)**

This approach consists of three main components: Product Line Mapping (PLM), Domain Potential Assessment (DPA), and Reuse Infrastructure Scoping (RIS).

In the product line mapping, a high-level description composed of the following elements is achieved: product line, domains, features, and their relationships. The PLM works in two main cycles. The first cycle is product-focused. At this point, the information referent to the product is systematized, the main features of the products are identified and their consistency is checked. The second cycle focuses on domains. In this cycle, features are grouped in terms of technical domains, i.e., cohesive groups of functionality. Additional features are also identified in this approach. However, this approach considers product scoping as a market science, and therefore, it does not completely address this type of scoping.

In the domain potential assessment, the benefits and risks associated to the various domains are identified and the sub-domains of each application domain are evaluated, identifying its reuse potential.

In the reuse infrastructure, a quantitative analysis of the benefits that can be expected from reuse is performed. In this phase, metrics based on Goal Question Metrics (GQM) (Basili et.al, 1994) are used to aid in the determination of the assets for reuse infrastructure.
A People Oriented Approach to Product Line Scoping: Enabling Stakeholder Cooperation with User Scenarios (Rommes, 2003)

In this work an approach that considers scoping not only as an economical activity, but also as a social activity is proposed. It considers that several stakeholders are influenced by the outcome of the scoping phase. Thus, to get an optimal result, as well as to gain commitment to this result, these stakeholders should be involved in the scoping phase.

For making possible that more stakeholders participate of the scoping phase, it proposes an approach centered on user scenarios. These user scenarios roughly define the product portfolio and are used as a basis for scope refinement, domain scoping, and architecting. However, only user scenarios are not sufficient for defining scope.

A Feature-Based Approach to Product Line Production Planning (Lee et al., 2004)

This study introduces a feature-based approach to product line production planning, and illustrates how feature models and feature binding information are used to identify core assets and develop a product line production plan.

In the definition of the planning, technical aspects, such as the assets which will make part of the reference architecture are considered.

QFD-PPP: Product Line Portfolio Planning Using Quality Function Deployment (Helferich et al., 2005)

The QFD-PPP is focused on the definition of product scoping. In it, the QFD is used as a systematic although more informal way of communication between customers and developers.

The work demonstrates how QFD-PPP can be used to identify different customer groups and their needs, derive a product portfolio (i.e. members of a product line) systematically and derive common and variable product functions including requirements that the customers would not have come up with. However, it is not possible to affirm that this approach is really effective, because its validation in the industry is still lacking.
This work presents an approach for domain analysis and economical evaluation of core asset scoping. The approach is composed of five activities, and each activity offers work instructions. This approach is used to analyze the domain, decide which features should be modeled into the core assets, and evaluate the economical value. Thus, it is possible to identify the core assets that provide the most economical value. However, this study does not deal with all abovementioned three types of scoping. It only proposes an approach for domain analysis and economical analysis of core asset scope.

Software Product Lines – SEI Framework (Clements, 2005)
The Software Engineering Institute (SEI) describes scoping as a technical management practice area in their general product line practice framework. They identify different practices such as: applying What to Build Pattern; examining existing products; conducting a workshop to understand product line goals and products; context diagramming; developing an attribute/product matrix; and developing product line scenarios. But, it does not define specific activities for scope definition.

A Practical Guide to Product Line Scoping (John et al., 2006)
This approach is an update of the PuLSE Eco (Schmid, 2002) and describes the generic scoping approach, which is the basis for particular applications of PuLSE Eco in industry. The update focuses on the concretization of customization characteristics needed for different product line settings.

With the application of this approach, the scoping can change according to the context. The general PuLSE categories for customization factors that are used in this approach are: operational context; domain characteristics; integratable artifacts; enterprise context; and resources.

This update of the PuLSE Eco, however, does not present a framework, which indicates a set of specific aspects for different projects.
Core Asset Scoping Method: Product Line Positioning Based on Levels of Coverage and Consistency (Inoki and Fukazawa, 2007)

This approach aims to identify the core assets of the product line analyzing the assets quality using metrics for consistence and coverage. The metrics of coverage measure the quality of the core assets and the consistence metrics is a quantitative measure that describes to what degree elements of core assets are prepared.

The main goals of this approach are: to solve problems related to the lack of evaluation of the software product lines according to the domain and quality of the core assets, to aid the enterprise in the planning of its investments in product lines, as well as to contribute for the optimization of software development focused on product lines.

Agile Product Line Planning: A Collaborative Approach and a Case Study (Noor et al., 2007)

This approach uses the PuLSE Eco approach as basis for its scoping activities. During the approach, tasks of Collaboration Engineering and practices of agile methods are used in parallel.

The main purpose of this approach is to define a product map, i.e. a matrix that relates products and features across the active participation and collaboration among the project stakeholders. This research area needs to be deeply explored to make possible the understanding of how to maximize the benefits that can be obtained with agile and SPL together, mainly during the scoping phase. Moreover, this approach is more associated to collaboration engineering tasks than to agile practices.

3.2.2.6. Data Analysis

The data analysis aims to identify the information about the approaches.

According to the previous analysis of the approaches, it was possible to identify:

- Limitations in scope definition of the current SPL approaches;
- Challenges in the area to be addressed; and
- Best practices of scope definition for SPL.
After analysis, the data were synthesized. The data synthesis involves assembling and summarizing the results of the included primary studies. For a better documentation of the approaches analysis, a form was created to keep it, besides the sub-questions and the available information about them.

### 3.2.2.7. Data Extraction

The data extraction forms must be designed to collect all the information needed to address the review questions and the quality criteria of the study.

The followings information were extracted from each study:

1. Study title and authors;
2. Sources (conferences, journals, and so on.);
3. Year when the study was published;
4. Classification (related work, such as survey, or approaches);
5. Research question answers;
6. Summary (a brief analysis of the approaches, overview of its weaknesses, strengths and characteristics); and
7. Quality criteria and critical analysis of the approaches.

Each study was documented following a pattern template, where all extracted data were documented, reviewer's name and date of the review. The Appendix B presents the template used for the documentation of the studies.

### 3.2.3. Reporting

In this section, the results of the review will be presented and each research question will be discussed and analyzed, and conclusions will be drawn.

### 3.2.3.1. Scoping Activities

Analyzing the primary studies, we found a wide variety of activities for scoping. However, the lack of standardization can be mentioned as a problem. In general, each approach analyzed has specific activities to define scope. Furthermore, the majority of the approaches do not have clearly identifiable activities that could be easily applicable in practice. Also the guidelines for their application are described in high level. Therefore, in general, the use of the approaches is considered difficult by organizations and their activities, typically, are adapted for the scope definition to have success.
In summary, the activities defined by the approaches can be clustered in three groups: *identification*, *analysis* and *prioritization*. Identification is related to features, requirements and products, the analysis is based on domains and sub-domains and the prioritization is related to optimization factors. However, there are few approaches which address activities related to scope optimization, and other activities, such as market analysis and release planning. The current scenario shows that there are not approaches that present activities for all the contexts in which the product lines can be inserted, occurring inevitably an adaptation of the activities or the use of two or more approaches together to identify the scope adequately.

### 3.2.3.2. Scoping Optimization

According to Schmid (2002), it is important to distinguish scoping according to three goal levels: **identification of a scope**, i.e., simply writing down the scope; **evaluation of a scope**, i.e., to determine advantages and disadvantages of a particular scope; and, finally, **optimization of a scope**, optimization of the scope in terms of identifying what should be built for reuse, it aims at the optimization of the underlying concept (product line, domain, asset), as well as optimizing some sort of benefit.

In our review, it was possible to identify optimization in product portfolio scoping and asset scoping. The optimization in product portfolio scoping is performed, in general, to determine a feature set aligned with market and business goals. In assets scoping, the optimization activities are related, often, to the identification of the assets implementation effort and their potential for each product and for whole product line.

Optimization of SPL scope is still an aspect partially addressed. Few are the approaches that directly define activities for optimization, as activities related to components economic evaluation, where economic models are used, and optimization criteria, which use parameters, such as: market segments, customer viewpoint, product value and so on.
3.2.3.3. Scoping Types

Analyzing the scoping types, we can identify the focus of each approach. In general, as can be seen in Figure 3.1, the approach address product portfolio scoping and assets scoping. With the review, it was possible to identify that more than 36% of the approaches have activities related to product portfolio scoping, focusing on the identification of products and features based on market analysis and products analysis which are economically useful and beneficial to product line development. Besides, assets scoping is addressed by more than 63% of the approaches analyzed, focusing on the potential analysis of the assets, identifying efforts and costs of implementation of them for product line, and making analysis of variability dependencies.

On the other hand, there are few approaches which define domain scoping, approximately 27%, and these are based on activities of domain potential assessment, considering risks and benefits of the domains for the product line.

Based on the scope types addressed by the approaches, we can identify the levels in which they perform scoping and their goals (Moraes et al., 2009).

![Scoping Types](image)

**Figure 3.1 Percentage of the scoping types**
3.2.3.4. Stakeholders and Roles into Scoping

The participation of representative stakeholders with well-defined roles is very important for the scoping phase. With the effective presence of them, risks associated with SPL are minimized, for example, the risk of including products lacking some essential feature or with unnecessary features. However, the majority of the approaches do not have an effective definition of stakeholders and roles, even with the importance of this information for the success of scope definition. This makes difficult to identify which approaches have an appropriated team for scoping. Moreover, the choice of the stakeholders and roles in projects depends on a series of factors such as: resources, project type, complexity, and so on (Moraes et al., 2009). Nevertheless, roles such as: domain experts, with technical or marketing knowledge; scoping expert, to conduct the tasks of the approaches and identify constraints; end user or customer, to enable a customizable SPL; architects, for the phase assets scoping, with the function of defining the conflicts and impacts of reusable assets; developers, to determine the effort to implement the features; and manager, to define the organizational goals, are the most relevant roles in the context of SPL Scoping (Moraes et al., 2009).

Among the analyzed studies, stakeholders and/or roles are identified in five of them:

- In (Rommes, 2003), the following stakeholders are defined: manager, product managers, sales persons, customers, users, system architects, developers and suppliers;
- In (Clements, 2005), the following stakeholders are defined: manager, marketing, developers, users, testers, tool developers, technology researchers and domain experts;
- In (Helferich et al., 2005), the following stakeholders are defined: customer, engineer, developers, expert in QFD and architects;
- In (John et al., 2006), the following roles are defined: scoping expert, domain expert and product line manager and
- In (Noor et al., 2007), the following stakeholders are defined: senior manager, market analyst and sales persons, customers, software architects, developers and maintenance staff.
3.2.3.5. Cost Models and Metrics

Analyzing the eleven studies, we identify that the lack of metrics is an aspect in the majority of the approaches. Information as effort and costs for the product line are attributes not considered by the majority of the approaches and when are, in general, it is only made based on estimations from experienced stakeholders.

Among the analyzed studies that define metrics can be highlighted:

- **(Schmid, 2002)**, in this work business goal operationalization to define metrics is achieved, as, for example, metrics which determine the cost of implementation of a specific feature. It is performed based on the top-down process that was derived from the GQM process (Briand et al., 1996), (Schmid, 1999). In this approach, the high-level goals are operationalized by transforming it into a benefit metric which defines the benefit of a specific feature. In this approach, characterization metrics are also used for which the experts can easily provide values, because the benefit metrics are not fully operational. With this approach, it is possible to define a simplistic model to express the desired business goals as the result of properties of the features and products, in a way that can be determined by the experts;

- **(Park and Kim, 2005)**, in this work cost metrics are used to analyze economical value of the core assets. Besides, in the analysis of the scope economical value are considered the variability and also dependencies among it. With this approach, it is possible to identify the economic impact which each asset has on the SPL;

- **(John et al., 2006)**, in this work the quality models measure economic factors such as effort and risk. Thus, the assets that maximize the economical return can be prioritized; and

- **(Inoki and Fukazawa, 2007)**, this work uses metrics to evaluate levels of coverage and consistency. The level of coverage is a quantitative measure that describes to what degree elements of core assets are prepared. It considers returns on investment and should be adjusted depending on the conditions of an
organization. This metric determines if an asset should be reused in various products. The level of consistency is a qualitative measure of all core assets, which describes the consistency of all core assets in a product line. Core assets with a high level of consistency do not contradict each other.

The use of metrics can be an effective way to optimize scope and to determine the costs associated with SPL. In this context, GQM has been shown as the solution more used by the organizations, because it makes possible to align the scope with the business goals.

### 3.2.3.6. Scoping Customization

John and Eisenbarth (2009) highlight that scoping, in practice, should be customized to concrete situation and activities, because representative stakeholders, artifacts and execution time for the tasks can change according to several factors, such as: resources, organizations factors, size of the team, domain characteristics, available assets and so on.

Among the approaches analyzed in this review, only the PuLSE Scoping Approach (John et al., 2006) defines customization factors which are adaptable for the context. These factors are grouped in five categories: 1. operational context, related to project and organizational constraints; 2. domain characteristics, related to domain complexity; 3. integratable artifacts, this category identifies the existence of artifacts relevant for scoping; 4. enterprise context, this category is related to the structure and maturity of the organization; 5. resources, related to the knowledge of the stakeholders and resources available for scope definition. However, this approach does not define a set of activities associated to each situation. The current scenario shows that a framework is necessary, in order to indicate specific activities, stakeholders and artifacts for different projects, driving the organizations in the application of scoping.

### 3.2.3.7. Agile Scoping of SPL

An important research area which is being investigated by the communities of product lines and agile is the join between SPL and Agile Methods (AM). Product lines and agile methodologies offer similar benefits, both aiming the
reduction of time-to-market and development costs. Nevertheless, these benefits are obtained in different ways (Noor et al., 2007).

According to (Noor et al., 2007), the union between SPL and AM can increase the potential of the organizations and open new markets for them. In this research area, there is only one approach related to scoping (Noor et al., 2007). This approach makes agile scoping, combining agile principles and practices, collaboration engineering and PuLSE Eco (Schmid, 2002). The goal of the approach is to define a product map which prioritizes features, domains and product with active participation and collaboration among the stakeholders. However, in general, this approach does not use all the potential which AM can offer, presenting as the most relevant agile characteristic the collaboration among the stakeholders. Finally, this approach was validated in only one industrial project and the lack of approaches which address agile aspects in SPL scoping make difficult to establish a comparison and evaluate the agility of the approach.

### 3.2.3.8. Scoping and SPL Development

The activities and results of scoping can have direct influence on the SPL development. However, the majority of the approaches do not offer well-defined support for relation between scoping and the development steps.

In general, the approaches define tasks for product lines started from scratch. Among the studies analyzed, only (Riebisch, 2001) clearly treat the results of scoping as basis for further development steps. In his study, three cases which relate scoping with the development stage of the SPL are discussed: 1. development from scratch; 2. development with existing products; and 3. evolution of an existing product line. In the three cases, it considers priority levels for features implementation, using decision tables. Considering the priority levels in which the features are and the development stage, the decision of implementing the features is performed.

### 3.3. Threats to Validity

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

The threats related to this review were:
1. **Research Questions**, it is possible that some questions defined in the protocol and which drive the systematic review are not so relevant. In order to avoid it, we had several discussions with RiSE members and experts in the area;

2. **Search of the Studies**, it is possible also that our search criteria did not find all the relevant approaches. Nevertheless, we searched the main conferences and journals and checked the references found in these papers to avoid that some relevant study was not selected; and

3. **Excluded Studies**, in this case, the research team had discussions, in order, to define a consensus about the work to be excluded. Moreover, criteria of exclusion and inclusion were defined to aid in this decision.

### 3.4. Empirical Studies

As cited in the Section 3.2.2.3. (**Studies Selection**) two surveys were found with a similar proposal (to evaluate the scoping approaches).

Schmid (Schmid, 2000) presented a survey of scoping-related technology, both from software engineering, as well as from non-software disciplines, presenting a framework for approaches analysis associated to two *problem dimensions*: task of scoping and object of scoping, and two *solution dimensions*: scoping product and scoping approaches. This framework is used to structure the field of scoping. The analysis made in (Schmid, 2000) had the following goals:

- Organize and structure the field of scoping;
- Provide an overview of what approaches exist for scoping and their relevant advantages and disadvantages;
- Aid in selecting among existing approaches; and
- Aid in improving existing methods and building new ones.

In (John and Eisenbarth, 2009) was performed the investigation of some issues in SPL scoping such as: the goal of the approaches, how to treat variability, the main inputs and outputs of the approaches, the involved roles, the effort to perform scoping, and the maturity and benefits with the use of the approaches.
Moreover, in this survey some new research questions were indicated:

- How is the connection of scoping and Requirements Engineering?
- How is the connection between scoping and architecture?
- How to produce quantifiable results?

It is important to highlight that our work performed the evaluation of the approaches using a systematic process. Thus, it can be replicated by other researchers and research groups, since all the process was documented and detailed in the protocol of the systematic review and can be accessed on the web page of this review\(^3\). In addition, several issues and studies that were not considered in (Schmid, 2000) as studies which define stakeholders, agile characteristics with SPL and so on were extensively discussed in our review. Moreover, our study can be useful to reinforce the findings identified in (John and Eisenbarth, 2009) since we investigated similar issues such as: main activities, inputs, outputs, defined stakeholders, and so on. On the other hand, our research complements their work since new issues are also investigated by our research questions, such as: customization, relationship between SPL and agile development principles and metrics.

3.5. Chapter Summary

Scoping is the initial phase of a SPL where the planning is performed. In it, the products and the features which will integrate the product line are defined and the costs and risks are identified.

As widely discussed along this chapter, the main purpose of this systematic review was to understand what is being considered by the existing SPL scoping approaches. Thus, the systematic review enabled that we identified the main aspects of SPL scoping, as well as the strongpoint and weakness of each approach. In Table 3.1 a summary with the analysis of the approaches, according to the main aspects of scoping, is presented.

\(^3\) http://www.cin.ufpe.br/~sople/scoping/sr/
### Table 3.1 Summary of the aspects of the scoping approaches

<table>
<thead>
<tr>
<th>Approaches Author</th>
<th>Activities</th>
<th>Optimization</th>
<th>Scope Types</th>
<th>Stakeholders</th>
<th>Metrics</th>
<th>Customization Factors</th>
<th>Agile</th>
<th>Relate clearly Scoping and Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Riebisch [2001]</td>
<td>identification, prioritization, interpretation of features</td>
<td>use priorities calculation and decision tables</td>
<td>Assets</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>Yes</td>
</tr>
<tr>
<td>Kish [2002]</td>
<td>identify requirements and products, list and prioritize architecture candidates, categorize requirements, examine candidates for the product-line scope</td>
<td>do not define optimization tasks for scoping</td>
<td>Product</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Schmid [2002]</td>
<td>identification and description of products, features and domains, product map prioritization, identification and prioritization of assets</td>
<td>optimize the feature matrix using GQM</td>
<td>domain, assets</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Rommes [2003]</td>
<td>writing and review of user scenarios, requirements identification</td>
<td>do not define optimization tasks for scoping</td>
<td>Product</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Lee [2004]</td>
<td>market analysis, feature modeling, feature binding analysis, assets identification, core assets development, production plan documentation</td>
<td>do not define optimization tasks for scoping</td>
<td>Assets</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Helferich [2005]</td>
<td>identify customer requirements, product functions and evaluate different technologies and architectures</td>
<td>use QFD to identify the true needs of the customers</td>
<td>Product</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Park [2005]</td>
<td>analysis of commonality and variability, variability dependencies analysis, domain model refinement, economical evaluation of core asset</td>
<td>do not define optimization tasks for scoping</td>
<td>Assets</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Clements [2005]</td>
<td>development of scenarios, attributes/products matrix, products analysis</td>
<td>do not define optimization tasks for scoping</td>
<td>Product</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>John [2006]</td>
<td>mandatory activities: product identification, plan product release, assess features, identify features, specify product feature matrix and identify domains</td>
<td>prioritize and optimized the feature matrix</td>
<td>domain, assets</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Inoki [2007]</td>
<td>product line development plans analysis, create product line development plan, create road map, define organizational standards</td>
<td>do not define optimization tasks for scoping</td>
<td>Assets</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>No</td>
</tr>
<tr>
<td>Noor [2007]</td>
<td>identify and agree domains, define features for each domain, analyze products, define products in terms of features, prioritize product map</td>
<td>make prioritization and optimization of the product map</td>
<td>domain, assets</td>
<td>yes</td>
<td>No</td>
<td>no</td>
<td>yes</td>
<td>No</td>
</tr>
</tbody>
</table>
As can be seen in Table 3.1, the majority of the approaches do not discuss the activities systematically and the guidelines for their applications are described in high-level. Other important issue is that the optimization of SPL scoping is partially addressed by the approaches, and there are not techniques or activities related to domain optimization. Moreover, in general, they do not consider the domains according to their relations with business goal and assets costs. Other finding was that among the approaches that define stakeholders’ roles, the main ones defined were: managers, customers, developers and architects. In addition, the lack of metrics is an aspect present in the majority of the approaches and only one of them defines customization factors for aligning scoping according to the context. Furthermore, only one approach considers the integration between SPL and AM, but it does not explore all potential which AM and SPL can have. Finally, we identified that the approaches define tasks for product lines started from scratch and do not treat clearly the results of scoping as basis for further development steps.

On the other hand, the main challenges identified are related to the following questions:

- How to maximize the potential of union between product lines and agile methodologies in scoping?
- How to customize scoping in SPL?
- Which techniques can be used to optimize scope?
- How the approaches combine scoping with the SPL development?

These questions have good potential of research and can be used as basis for new approaches of SPL scoping.

In addition, the effort and quality of the review present an important contribution that can be used as background information for scoping researches and companies that use SPL or are planning to adopt it, since it presents an important view of the state-of-the-art in scoping approaches, showing how scoping is addressed by the studies analyzed.

Next chapter presents an overview on agile software development and its relationship with software product lines.
Agile Software Development initially appeared based on the idea of iterative, incremental and evolutionary software development (Glyterud et al., 2008).

In 1930s were proposed shorts plan-do-study-act (PDSA) cycles for quality improvement. In 1940s several work were also performed with PDSA (Larman and Basili, 2003). In 1950s, the practice of iterative and incremental development was considered the main success factor of the project X-15 hypersonic jet (Dana, 1993). Other projects and research on iterative and incremental software development continued and in 1970s was defined the waterfall model by Royce (Royce, 1970). In 1970s, the term evolution and evolutionary was also introduced to a process (Gilb, 1976), where the evolutionary project management was discussed.

The Scrum method took inspiration from a Japanese iterative and incremental development approach used for non-software products at Honda, Canon, and Fujitsu in the 1980s (Takeuchi and Nonaka, 1986).

The spiral model was defined by Boehm (Boehm, 1986) and the Rational Unified Process emerged around 1990 from various studies (Larman and Basili, 2003). Further iterative, incremental methods such as eXtreme Programming (XP) and Feature-Driven Development (FDD) were developed during the 1990s and from 2000s through the present appeared methods such as: Lean, Evo, Crystal Family and so on.
In 2006, the agile paradigm was inserted in a new context: product lines. Thus, studies as Tian and Cooper (2006), Carbon et al. (2006), Ghanam and Maurer (2008), among others were published in this research area with the goal of investigating the benefits of the insertion of agile characteristics into product lines and of defining approaches with the utilization of the two paradigms.

In this sense, this chapter presents an overview on Agile Software Development, analyzing their benefits, limitations and main methods with techniques, principles and practices associated. Besides, it compares agile development and product lines in order to identify the viability of incorporating agile aspects into SPL and the main challenges from combining these approaches.

4.1. Introduction
The constant changes and the search for new benefits in the software industry make possible the emergence of new practices and research areas. In this context, a trend that has shown particular benefits is the area of agile software product lines.

The application of Agile Methods (AM) is characterized as a success practice in industry, where many small organizations have shown interest, because they seek alternatives to the traditional software development methodologies, which they find too cumbersome, bureaucratic, and inflexible (Lindvall et al., 2004). In the agile methodologies, the agility is seen as the ability to adapt quickly and flexibility to changes in processes, products and environments (Aoyama, 1998). In this sense, the methods for agile software development constitute a set of practices for software development that have been created by experienced practitioners [Gerfalk, 2006]. These methods can be seen as a reaction to plan-based or traditional methods, which emphasize “a rationalized, engineering-based approach” [Dyba, 2000; Nerur et al., 2005] in which it is claimed that problems are fully specifiable and that optimal and predictable solutions exist for every problem.

Agile development methods allow higher customer satisfaction and development processes that make possible developers to quickly create high quality software in business domains and technological environments that rapidly change with faster development time. AMs are also directed to the rapid
delivery of small features set in short interactions, where the software deliveries are the measurement of progress of the project. In this context, the focus is on coding, thus, planning and design are simple and performed in an incremental way.

On the other hand, another success practice in industry is the adoption of SPL, where the development of systems is based on domain knowledge, systematic planning, flexible and detailed design, reuse of core assets and exploiting commonalities in the domain. In this sense, the development of product lines requires highest time and efforts up-front to obtain representative results.

However, according to (Noor et al., 2007), in spite of the differences presented previously, the union of these two approaches can provide new benefits creating opportunities for new markets, balancing agility and plan-driven software development.

### 4.2. Agile Development – An Overview

In this section, an overview about agile software development is performed. In this overview, their general characteristics are discussed as well as their main methods.

#### 4.2.1. Agile Values and Principles

The agile development methods are based around four main ideas defined in the “Manifesto for Agile Software Development”. The Agile Manifesto was established in 2001 by a group that defined a set of values and principles to origin the agile paradigm.

The values defined by the Agile Manifesto are:

- Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation; and
- Responding to change over following a plan.

In order to satisfy these values some principles should be respected:

---

4 [http://www.agilemanifesto.org](http://www.agilemanifesto.org)
Chapter 4 - An overview on Agile Software Development and its Relationship with Software Product Lines

• Highest priority is to satisfy the customer through early and continuous delivery of valuable software;
• Best architectures, requirements, and designs emerge from self-organizing teams;
• Welcome changing requirements;
• Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale;
• Build projects around motivated individuals;
• At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly;
• Simplicity, the art of maximizing the amount of work not done, is essential;
• Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely;
• Working software is the primary measure of progress;
• Business people and developers must work together daily throughout the project;
• Continuous attention to technical excellence and good design enhances agility; and
• Most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

4.2.2. Agile Development Methods

The argument for the choice of agile development methods should be based on aspects related to size of project and team, time constraints and project domain. AM are generally used in small projects, with small teams, where there are time limitations and higher volatility in the domains. According to Boehm (2002), agile methodologies are difficult to scale up to large projects because of the lack of sufficient architecture planning, over-focusing on early results and low levels of test coverage.

In the literature, various methods propose agility in their definitions, aiming to find efficient ways for developing software of quality across an agile
development process. These methods focus on values and principles defined by the Agile Manifesto (See Section 4.1.1), as can be seen in the next sections.

In Table 4.1, the main agile development methods are summarized and briefly described.

**Table 4.1 Summary of agile methods**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XP</strong></td>
<td>Consists of a set of practices such as pair programming, continuous integration and simple design, that has to be followed to be agile and has a lifecycle with four basic activities: coding, testing, listening, and designing.</td>
</tr>
<tr>
<td><strong>Scrum</strong></td>
<td>It aims to manage and control the software production using iterative, incremental and lightweight processes (Schwaber and Beedle, 2001)</td>
</tr>
<tr>
<td><strong>Crystal</strong></td>
<td>It is a family of methodologies with a common genetic code, one that emphasizes frequent delivery, close communication and reflective improvement (Cockburn, 2004)</td>
</tr>
<tr>
<td><strong>FDD</strong></td>
<td>Provides a combination of model-driven and agile development. An object model is produced first, then features of the model are established and a plan is devised based on these features. Then the design and build are performed iterative for each feature (Gylterud et al., 2008).</td>
</tr>
<tr>
<td><strong>DSDM</strong></td>
<td>DSDM provides a framework of controls and best practice for Rapid Application Development (RAD) and is particularly suitable for application development projects that need to develop complex business solutions within tight timeframes (Hunt, 2006)</td>
</tr>
</tbody>
</table>

**4.2.2.1. eXtreme Programming (XP)**

XP is a style of software development focused on industrial application of programming techniques, clear communication, keeping and small development teams. Furthermore, researches of XP describe that this method has advantages when compared with others traditional processes, such as higher customer satisfaction and team productivity and shorter release cycles.

The XP lifecycle has four basic activities: *coding*, *testing*, *listening*, and *designing*. Moreover, it encourages continual communications with customers and teams, maintaining simplicity, providing frequent feedback via testing, and dealing with problems proactively (Paulk, 2001). XP also contains several
practices that support tacit group knowledge and are used to achieve their principles.

The practices of XP focus on the following points (Fowler, 2000; Loftus and Ratcliffe, 2000; Beck and Andres, 2004; Hunt, 2006):

- **The Planning Game**: A series of planning activities, that XP calls *planning games*, are undertaken during the course of a project. An *Initial Planning Game* is undertaken during the project inception and involves the customer in a brainstorming session with a few key personnel from the developer company to determine feasibility and outline key initial requirements, known in XP as *stories*. If the project goes ahead, a *Release Planning Game* is undertaken where the customer writes further *stories* on index cards and developers estimate effort in terms of Ideal Engineering Units. The customer also plans a development iteration schedule — each iteration running for between two and four weeks — and shares out *stories* across these iterations on the basis of business priority. At the beginning of each iteration, an *Iteration Planning Game* is undertaken where the customer can add further *stories*, and change the *story* schedule.

- **Code in pairs**: Pair programming is a dialog among two people simultaneously programming and trying to program better. The concept is that if code reviews are good, because they force at least one other person to consider the code, then constantly working on pairs results in constant reviewing of code and feedback among the two developers.

- **Merciless refactoring**: Developers are required to improve code without changing its functionality in order to produce the simplest possible code that will work.

- **Apply coding standards**: The code is the main form of documentation. Therefore, it must be clear and written in a consistent fashion.

- **40 hour week**: The aim is not to overwork employees but allow them to maintain a good balance between work and life.
Chapter 4 - An overview on Agile Software Development and its Relationship with Software Product Lines

- **Continuous integration:** Code that passes unit tests should be integrated with the system at least once a day. This allows developers to detect system integration bugs at an early stage by running the project’s suite of unit tests.

- **System metaphor:** A language of metaphors is used to express the architecture of the system. This aids communication among developers and between developers and the customer.

- **Stay in contact with the customer:** In this case, the presence of a representative customer in the team is very important in order to make easy the contact between the team and the customer.

- **Test-first programming:** Developers should write the unit tests before the code to be tested. The argument for this is that if they cannot write the test, then they are not able to write the code.

- **Short iterations:** Each iteration should be relatively short allowing rapid and frequent feedback.

- **Incremental design:** This indicates to invest in the design of the system every day. Striving to make the design of the system an excellent fit for the needs of the system that day.

- **Collective ownership:** Everyone within the team owns the software and has responsibility for it. XP does not support a culture of blame and recrimination – everyone is responsible for all the code. As a result of this, everyone is responsible for fixing a problem when they find it rather than ignoring it.

Based on these principles is possible to affirm that XP makes possible for their developers a collaboration and a sense of project control. By adopting XP, developers know where their project is heading and whether it is delayed (Cao et al., 2004).

With the overview was possible to identify that XP is extensively defunded between researchers and organizations, and that the XP projects are characterized for small teams in environments co-located or distributed, with projects of small and medium port, in which simplicity, collaboration and cooperation make part of the business culture. However, in spite of offering benefits such as customer satisfaction and fast response to change requests, in
large projects the benefits cited are pledged and the utilization of XP is not recommended. Furthermore, the practices in XP are rarely applicable (Keefer, 2003) making difficult the success with the application of this method. Among the practices of XP, the practice of pair programming is supported by a more well-established stream of research (Erickson et al., 2005). Other limitation is related the maturity of the development teams, because few studies on XP were performed on mature development teams (Dyba and Dingsoyr, 2008).

4.2.2.2. SCRUM

Scrum is a development method applicable for volatiles environments. It has as main characteristic the focus on the management, aiming to manage and control the software production using iterative, incremental and lightweight processes.

Scrum is based on current process control theories and specifically aims to produce the best end results, given the current resources and time available (Hunt, 2006). This agile development method has as basic aspects the following principles: small team, small iterations (sprints), unstable requirements, feedbacks and monitoring.

In Scrum, the teams are composed of the following roles (Schwaber, 2004):

- **Product owner**: Representing the interest of all in the project. It defines the fundamentals of the project and prioritizes the requirements which will be implemented in each sprint, so, the requirements of more value can be first implemented.

- **Scrum master**: Managing the process of the Scrum, assuring who all in the project perform the process according to their rules.

- **Team**: Developing the functionalities of the products, being collectively responsible for the success of the project.

In the method, the project flow starts with the identification of the *vision* of the product. The *vision* is a document in which are related the products’ issues defined by clients, some premises and constraints. After the definition of the vision, a list, called *product backlog*, composed of all requirements
Chapter 4 - An overview on Agile Software Development and its Relationship with Software Product Lines

identified to the product is created. In Figure 4.1 the flow of the Scrum can be seen in more details.

Figure 4.1 Scrum. Adapted of Hunt (Hunt, 2006)

In Scrum, the requirements development is performed in sprints, in which the planning of “what” should be implemented is defined. In this method, the work of architecture and infrastructure are achieved in the first sprints, and each sprint has duration from thirty days (maximum).

In the execution of the sprints, the project team daily makes a meet (Daily Scrum Meeting) with duration of fifteen minutes with the objective of verifying the work progress. After each sprint, a Sprint Review Meeting is performed, in which the outcomes obtained in the sprint are shown for the product owner. After the Sprint Review Meeting, the scrum master conducts the Sprint Retrospective Meeting, with the objective of improving the process, the time and/or product. The Sprint Retrospective Meeting is the last activity performed in each sprint, thus, with its conclusion a new sprint can be planned and executed.

With the Scrum analysis, is visible their bigger preoccupation with project management. Further, it provides values and procedures that can help to introduce a more dynamic and responsive development process. However, as
well as in XP, Scrum presents some limitations such as: it should not be implanted in large projects and needs more scientific studies (Dyba and Dingsoyr, 2008).

4.2.2.3. Crystal

Crystal is a family of methodologies that emphasizes frequent delivery, close communication and reflective improvement (Cockburn, 2004). There is no one Crystal methodology. There are different Crystal methodologies for different types of projects, where each project or organization uses different properties to generate new family members (Cockburn, 2004).

Crystal is based on three properties (Cockburn, 2004): 1. safety in the project outcome; 2. efficiency in development; and 3. habitability of the conventions. Besides, according to Cockburn (2004), Crystal has the project team steer toward seven safety properties. These properties are: 1. frequent delivery; 2. improvement; 3. active communication; 4. personal safety; 5. focus; 6. easy access to expert user; and 7. frequent integration.

In Crystal, the amount of details needed in the requirements, design, and planning documents varies with the project circumstances, specifically the extent of damage that might be caused by undetected defects and the frequency of personal collaboration enjoyed by the team. Crystal defines four main roles. These are (Cockburn, 2004):

- **Sponsor:** This is the person who is either allocating or defending the allocation of the money for the project. The executive sponsor is supposed to keep the long-term view in mind, balancing the short-term priorities with those of subsequent releases and subsequent teams evolving or maintaining the system. This is the person who will create outside visibility for the project and provide the team with crucial business-level decisions.

- **Expert user:** This is the person who is supposed to be familiar with the operational procedures and the system in use (if there already is one), knowing which are frequently and infrequently used modes of operation, what short-cuts are needed, and what
information needs to be visible together on the screen at the same time.

- **Lead designer:** This is the lead technical person, the person supposed to have experience with software development, able to do the major system design, tell when the project team is on-track or off-track, and if off-track, how to get back onto track.

- **Programmer:** This is a person who is responsible for codification, tests and domain models.

Crystal is composed of a set of techniques that provide strategies characterized as a good start point for agile development.

The *Methodology Shaping* gathers information about prior experiences and uses it to come up with the starter conventions. It is performed in two steps: 1. project interviews, they make possible a library of experiences in their organization that shows strengths and weaknesses of the organization and 2. methodology shaping workshop, it is used for periodic reflection of the team.

The technique *Reflection Workshop* says that the team should pause for an hour periodically after each delivery. In the reflection workshop, the team members discuss what is working well, what needs improvement, and what they are going to do differently during the next period.

The *Blitz Planning* is a technique related to the project planning and has a collaborative nature. In the *Blitz Planning* the project map and timeline are defined. This technique is similar at the *Game Planning* from XP.

The *Delphi Estimation Using Expertise Rankings* is a technique that aims to estimate the size of the system to be built, the work time, identifying suggesting releases, by technical and business dependency and balancing the releases into approximately similar sizes.

The *Daily Stand-up Meetings* are shorts meeting to trade notes on status, progress, and problems. The meeting is not used to discuss problems, but to identify problems.

The technique *Essential Interaction Design* aims to produce shared understanding among the sponsors, users and developers and paper-and-pen collages ("reminding markers" in cooperative-game language) showing roles, tasks, and interaction contexts.
The *Process Miniature* is a technique used to reduce the problem that people have with starting to use an unfamiliar process, making possible a general vision of the process.

*Side-by-Side Programming* aims to decrease the negative effects of pair programming. In Side-by-Side programming, two people sit close enough to see each others' screens easily, but work on their own assignments.

*Burn charts* have become a favorite way to give visibility into a project's progress. They are extremely simple and astonishingly powerful. They reveal the strategy being used, show the progress made against predictions, and open the door to discussions about how best to proceed, including the difficult discussions related to decrease the scope or extend the schedule.

With the definition of these techniques, Crystal enables a vast group of possibilities for that organizations adopt the agile strategies most suitable for their projects. Nevertheless, Crystal does not support distributed development (Qumer and Henderson-Sellers, 2007) and lack scientific studies related to its uses and evaluation (Dyba and Dingsoyr, 2008).

### 4.2.2.4. Feature-Driven Development (FDD)

FDD combines Model-Driven Development (MDD) and agile development with emphasis on initial object model, division of work in features, and iterative design for each feature (Palmer and Felsing, 2002). Further, this method provides a way of controlling the iterative and incremental nature of agile projects (Hunt, 2006). In Figure 4.2 the flow of FDD projects is shown.

FDD is a method centric on the client, in the architecture, and besides, is a pragmatic method. Initially, an overall model is developed and next a features list is built. After this step is performed the project planning that is based around timeboxes (time interval) that are used to define the size of each interaction. Timeboxing iteration has a number of benefits including the ability to schedule and plan for incremental releases of the software, schedule and implement features, manage budgets and monitor progress within fixed time constraints.
In the planning of FDD projects, the features are associated with a cost, a priority and resources, where costs and resources can be determined by examining the number of person days taken to accomplish the feature and the priority can be determined by the following factors: architectural importance of the feature, utility for the users, risks involved and requirements of the system. In FDD, the design and the construction are also feature-driven.

In design phase, the impact of the features on the architecture is analyzed and new architecturally significant entities can be identified. Moreover, this phase can involve optionally the review of the architecture. In FDD, the architecture does not try to be all encompassing and incorporate hooks for all possible required features, it should represent the core features or those features that are most likely to be incorporated.

The development in FDD is performed in a managed and incremental way. In this phase, the features are implemented and after the implementation, unit and acceptance tests are executed. All the tests should pass for that the system be deployed to the client.

FDD is an agile methodology that clearly presents a greater emphasis on planning and aims adaptive, iterative and agile projects. It breaks the paradigm that considers agile methodologies as approaches totally focused on the implementation and enables a planning totally managed.
4.2.2.5. Dynamic Systems Development Method (DSDM)

DSDM provides a framework of controls and best practices for Rapid Application Development (RAD) and is particularly suitable for application development projects that need to develop complex business solutions within tight timeframes (Hunt, 2006). These solutions are guided by nine principles of highest importance (Stapleton, 2003):

1. user involvement;
2. empowering the project team;
3. frequent delivery;
4. addressing current business needs;
5. iterative and incremental development;
6. allow for reversing changes;
7. high-level scope being fixed before project starts;
8. testing throughout the lifecycle; and
9. efficient and effective communication.

In DSDM, the projects are divided in three phases: pre-project, project lifecycle, and pos-project. The pre-project enables that the users and the team understand the application as well as what will be expected for the end product. The project lifecycle is the phase that starts with the feasibility analysis of the use of DSDM to the particular project. After the feasibility analysis, a business study is performed, high-level requirements are identified, the architecture is defined and a prototyping plan is defined to be adopted for the development of the end product. With the end of the business study, the features to be designed and implemented are analyzed and a functional model and prototypes are developed to help improve the analysis and understanding of the application. In the project lifecycle is also performed the build and implementation of the application. The build involves code creation to implement the required functionalities and test of the application. On the other hand, implementation involves transfer the application for the production environment. After the end product delivery, the activities of pos-project are started. These involve activities of benefits for the users, such as, help desk.

DSDM has as main aspects facilitated workshops, prioritization, timeboxing and prototyping, focusing also on the risk management, configuration management and modeling.

4.2.3. Benefits

In general, an AM presents techniques applicable in a simple way. Furthermore, agile software development methods have the ability to respond to changes in customer requirements and needs throughout the development process. Agile
methods handle changing requirements by using iterative development, where the requirements evolve during the start of each cycle (Abrahamsson et al., 2002; Cohen et al., 2004). This fact makes Agile Software Development methods able to respond to changes in customer needs and implement the needed functionality instead of producing unnecessary functions.

Other benefits related to the agile software development are the quality, the low costs and the decrease of time-to-market. The quality is the main aspect investigated not only by the agile methodologies, but also by all development methodologies. With simple and incremental planning and design, an AM enables low cost of project and decreases time-to-market. Another factor associated to the decrease of time-to-market is the delivery of working software in each iteration, so, in theory, the software could be delivered and used by the customer before the whole software is finished, thus reducing time-to-market (Gylterud et al., 2008).

AM also enables the highest satisfaction of the customer. For this, the role of the customer is imperative and constant feedbacks are obtained of them. The effective participation of the customer in the project provides simplicity in the relationship between the team and customer decreasing the needs of rigid contracts.

4.2.4. Limitations
In spite of the benefits provided by the adoption of agile methodologies, the studies that address the introduction and adoption of agile methods do not provide a unified view of current practice (Dyba and Dingsøyr, 2008). Moreover, it is clear the lack of studies related to non-well known methods, such as Crystal, DSDM, FDD and so on. In this sense, there is little scientific support for many of the claims made by the agile community (McBreen, 2003).

Another limitation is the lack of attention for the design and architectural analysis (Stephens, 2003). Further, studies affirm that agile development methods are not appropriated for large projects and are more suitable for small teams (Cohen et al., 2004).
4.3. Incorporating Agile Aspects into Product Lines

The current market expectations force enterprises to produce new products with highest quality in short time. In this sense, product lines and agile development are characterized as success practices in industry.

Nevertheless, the essence of the two approaches is different. Agile development presents direct customer collaboration, simple planning and short interactions. On the other hand, product lines have indirect customer collaboration using well-trained customer representatives, are plan-driven and need of a scoping up-front to obtain success. However, in spite of apparently incompatibles, both approaches present similar benefits: decrease time-to-market and increase quality of the products, where decrease time-to-market is obtained in product lines by delivering customized software with reuse, and in agile development across incremental software development. Quality increase is obtained by continuously integrating code and detecting defects earlier in both approaches (Gylterud et al., 2008).

4.3.1. The Challenges

Comparing SPL and AM some differences are evident (See Table 4.2), generating various challenges for the development of Agile Software Product Lines, such as planning, documentation and architecture.

Product lines require high effort and costs up-front with scoping, design and core assets development. On the other hand, agile development methods present low cost and efforts, characteristics possible by the simplicity implanted in these methods. Costs reduction and efforts to develop a product line is a challenge and, in many cases, the high initial investment discourages the adoption of SPL by enterprises.

In SPL, it is clear the importance of well-defined processes and documentation. Successful scoping must target the right products for the domain, determined by factors such as knowledge of similar systems and future market demands, and it must have the appropriate scope, because too large or small scope will affect the capability of the SPL to achieve the variability and/or economies (Clements and Northrop, 2001; Tian and Cooper, 2006). There is also the need to plan for future projects that are potential members of the product family. This aspect is a challenge for AM, where the requirements are
defined in high-level and the planning is simple and focused on the needs of each iteration, having as main documentation the code.

Another challenge is in the architectural planning. SPL requires flexible architecture, the key for the development of customized applications. The flexibility of the architecture has as objective to accommodate variations in the product line, making possible the customization. On the other hand, agile methodologies emphasize the simplicity to minimize the amount of work. Thus, architectural aspects that are not of immediate interest for the current iteration are not considered in the architectural decisions.

A characteristic also investigated in this work was the formation of project teams and the role of the customers in these teams. In general, agile projects are composed of small teams with active interaction of the customer. In AM the role of the customer is critical, while in SPL, it is just important and the project teams are composed of roles well-defined, divided between domain and application. Table 4.2 compares SPL and AM with a summary of the aspects discussed in this section.

**Table 4.2 Comparing SPL and AM (Qumer and Henderson-Sellers, 2007; Hanssen and Fægri 2008)**

<table>
<thead>
<tr>
<th></th>
<th>Software Product Lines</th>
<th>Agile Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Planning</strong></td>
<td>There is a need to plan for future projects that are potential members of the product family. Requiring high investment up-front</td>
<td>Simplicity in the planning. Requirements are initially defined in high level</td>
</tr>
<tr>
<td><strong>Team's size</strong></td>
<td>Large</td>
<td>Small and Medium</td>
</tr>
<tr>
<td><strong>Customer's role</strong></td>
<td>Important</td>
<td>Critical</td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td>Scope centric</td>
<td>People centric</td>
</tr>
<tr>
<td><strong>Documentation</strong></td>
<td>Well-defined documentation</td>
<td>In general, there is not well-defined documentation. The main documentation is the code</td>
</tr>
<tr>
<td><strong>Architecture</strong></td>
<td>Dynamic and flexible architecture</td>
<td>Simple architecture. Architectural aspects that are not of immediate interest for the current iteration are not considered in the architectural decisions</td>
</tr>
<tr>
<td><strong>Development</strong></td>
<td>Composed of two cycles: core asset development and product development</td>
<td>Iterative and incremental</td>
</tr>
</tbody>
</table>
4.3.2. The Viability
Carbon et al. (2006) believe that the two approaches can be combined. Product lines as a proactive, strategic reuse approach used as basis to develop new products in short time and with high quality and agile methods used in application engineering to perform the customization of a product for a specific customer. In this work, they show how agile principles can be incorporated into application engineering using Fraunhofer PuLSE-I approach.

Tian and Cooper (2006) compare AM and SPL from different perspectives and conclude that is difficult to join agile methodologies with a plan-driven approach.

Noor et al. (2007) define an approach that balances agility and product lines planning with the aid of Collaboration Engineering. The goal of the approach is to define a product map which is prioritized in term of features, domains, and product using as basis the Fraunhofer PuLSE-Eco approach.

Ghanam and Maurer (2008) discuss the challenges of constructing agile product lines and propose to build a bridge between AM and SPL to combine the advantageous characteristics of both.

Gylterud et al. (2008) identify several challenges in combining product lines and agile development, among them are: requirements, design and creativity. Besides, they conclude that one possible solution for Agile Software Product Lines is test-driven development approaches.

Hanssen and Fægri (2008) defined a case study to understand how product lines and agile could be combined and concluded that in fact these two approaches complement each other with SPL supporting strategic objectives and AM tactical objectives.

Thus, considering the aspects shown in these work, we believe that the construction of new approaches involving the paradigms of product lines and agile methodologies is viable and highly beneficial. For this, it is important to combine the advantageous characteristics of each one and investigating “which” agile characteristics can be adapted at the context of product lines and “how” the adaptations can be performed.

4.4. Chapter Summary
In this chapter, the main agile development methods and the relationship
between AM and SPL were discussed.

Initially, characteristics such as agile principles and values were presented as well as general aspects of well know agile methodologies. In addition, the benefits and limitations about AM were also discussed. The main goal with this analysis was to identify, among the relevant methods, characteristics which can be combined with product lines and offer directions on how to make it.

For identifying the viability and the challenges of the join between SPL and AM, an initial base for comparison was defined. Based on it, we identified that this join is viable and highly beneficial. Thus, we develop an agile scoping process for product lines, aiming to join the potential of SPL and AM. This process is described in the next chapter.
Scoping has been identified as an important phase of the lifecycle of software product lines because it is directly associated to the economical success of them (Clements, 2002). It constitutes the initial phase of product lines and has as goal to determine the viability of the product line and maximize the economical benefits obtained with their development.

This chapter presents an agile scoping process for software product lines called RiPLE-SC. RiPLE-SC is part of the RiPLE (The RiSE Process for Product Lines Engineering), process that fully covers the product lines lifecycle, involving scoping (RiPLE-SC), requirements engineering (RiPLE-RE), design (RiPLE-DE), implementation (RiDE) and tests (RiPLE-TE). As can be seen in Figure 5.1, the RiPLE covers still risks management across the RiPLE-RM and evolution management across the RiPLE-EM.
5.1. Introduction
In software product lines engineering, scoping was first profoundly introduced by Debaud and Schmid (1999) as “a critical activity because it elicits the infrastructure from which family members can be common realms upon which the different products of a product line can be optimally engineered with respect to economies of scope”. However, neither Debaud and Schmid’s work nor many other work (Clements and Northrop, 2001), (Jhon et al., 2006), (Helferich et al., 2005) address the issue “how to perform systematic and agile scoping”.

The importance of scoping to be systematic in SPL is associated to the characteristic that it should manage variabilities and commonalities among several applications, factor which increases its complexity compared to traditional software development. Thus, to develop a SPL, it is needed that systematic scoping been performed to obtain positive results. On the other hand, as SPL scoping is totally defined up-front, it demands high effort and costs. In this sense, introducing agile aspects such as iterative and incremental, collaborative work among all the stakeholders of project, communication face-to-face, timeboxes and so on, in SPL scoping enables to add agility for scoping and consequently to decrease effort and costs.

In this context, a systematic and agile scoping process with techniques, tasks, guidelines, inputs, outputs and roles well-defined incorporating agile practices in their lifecycle is important to decrease the risks related to product lines scoping and decrease the effort and costs up-front.

The union between product lines and agile methodologies is a current research tendency, where the focus is on searching new combinations of practices in order to provide new benefits (McGregor, 2008). According to McGregor (2008), three dimensions are of interest in characterizing the business environment regarding the suitability of an agile product line process:

- The degree of commonality among the products, which determines the percentage of reuse of the products;
- The volatility of the relevant domains, which determines the degree of changes of the domain; and
- The magnitude of the project, i.e., the size of products, teams, and the organization.
According to Ghanam (2008), there is a need of investigating whether and how AM and SPL can work together to achieve their common goals. Moreover, Noor (2008) affirms that the union between product lines and agile process can increase the potential of the organizations and open new markets for them. In this context, was defined the RiPLE-SC that has as main purpose to produce scoping with simplicity and few efforts and costs upfront, maximizing the economical return obtained with the product line. Thus, in order to obtain a better result of scoping, a systematic and agile process, which uses the potentials of product lines and agile methods, is essential to provide guidance for the project team, specify direct tasks to be executed and guidelines to make possible the agility.

5.2. The Principles
In order to provide a systematic way to perform agile scoping, the proposed process is based on a set of product line principles combined with agile ones. The principles used by the RiPLE-SC are discussed next.

5.2.1. SPL Principles
In this section, the principles of the RiPLE-SC associated to product lines are elicited and discussed.

5.2.1.1. Metrics
The application of metrics in scoping can be a form to optimize it and defines the reusable assets of the product lines. The optimization of scoping is associated to the adaptation of the results of scoping with specifics objectives (for example, features of products can be inserted or not at the final scope according to the effort to develop them). In this sense, the RiPLE-SC uses The GQM for deriving metrics from business goals. These goals are defined by the stakeholders of the product line, enabling them to obtain a scope aligned with the business goals identified with base on their perspectives for the project.

5.2.1.2. Features Oriented
As agile approaches, such as ASD (Highsmith, 2002) and FDD (Hunt, 2006), RiPLE-SC is feature-oriented. Features are distinct and prominent concepts or
issues of a domain, and are externally visible at the users or others stakeholders (for example, analysts, designer, developers, etc) (Lee et al., 2002).

The concept of features is related to the problem domain if characterizing as an intuitive form of the people express commonality and variability of a domain (Lee and Muthing, 2006).

RiPLE-SC considers a feature of two different kinds: non-functional features, which represent quality attributes and constraints associated to products; and functional features, which represent characteristics visible for customers. The choice of a feature-oriented process makes possible the best understanding and communication between customers and developers. Moreover, given the observation that products and domains can be characterized in terms of their features, it was a rather straightforward decision to base our process on the notion of features.

5.2.1.3. Commonality and Variability Analysis
Software product line engineering explores the commonalities and variability among a set of related or family of products (Clements and Northrop, 2001). The goal of commonality analysis is to identify which features are common to all products of the domain. On the other hand, the goal of the variability analysis is to identify which features differ among the products, and determine precisely the differences.

5.2.1.4. Coupling
Coupling is a concept that directly impact on the choice of the domains. It identifies if the domains can be dealt in an independent way.

In this context, the RiPLE-SC considers coupling as an investigation dimension for the domain analysis in the phase of domain scoping (See Section 5.4.2).

5.2.1.5. Systematic Sequence of Activities
The last principle of product lines, and not less important, is to provide a systematic sequence of activities, which are divided into a set of tasks with well-defined inputs and outputs, and performed by a predefined set of roles with clear responsibilities. The purpose of this systematization is to easy the use and
adoption of the RiPLE-SC in practice and, consequently, increase the agility of the process use.

5.2.2. Agile Principles
In this section, the principles of the RiPLE-SC associated to agile development methods are discussed.

5.2.2.1. Iterative and Incremental
A process is considered iterative by the act of repeating a process usually with the aim of approaching a desired goal, target or result. Each repetition of the process is also called “iteration”, and the results of each iteration, the incremental part of the process, are used as the starting point for the next iteration (Jacobson et al., 1999).

In this sense, RiPLE-SC searches to define the scope in an evolutionary way, enabling various cycles of feedback and the refinement of the scope during the lifecycle of the SPL. Moreover, software systems are very complex to will be totally planed up-front.

5.2.2.2. Satisfy the Customer
The satisfaction of the customer, in general, is an aspect constantly searched in AM and SPL. In the RiPLE-SC, customers have an important role. Across the active participations of the customers, it is possible to identify their real needs and obtain constant feedbacks.

5.2.2.3. Communication Face-to-Face, Collaborative Participation, Simplicity and Profile of the Team
According to affirms the Agile Manifesto, the communication face-to-face is an efficient and effective method of communication among the project team. Each member of the team should be individually motivated and maintain a collaborative relationship, making possible that business people and technical people work together daily.

Other characteristic that influence in the agility of the processes is the technical excellence of the team. Therefore, RiPLE-SC proposes activities performed in the form of workshops, which enable the communication face-to-
face and the collaborative relationship between technical people and business people.

The workshops also enable to align the knowledge of the team and, consequently, decrease the production of documentation, favoring the simplicity.

### 5.2.2.4. Constant Feedback

The process proposes to use validation meeting for making possible constant feedback and increase the quality of scoping.

### 5.2.2.5. Team’s Reflection

The process proposes that at regular intervals, the team can reflect on how to become more effective, then tunes and adjusts its behavior accordingly.

### 5.3. RiPLE-SC Overview

RiPLE-SC is an agile and systematic process for scoping. It is the first discipline of the RiPLE, as can be seen in Figure 5.2, and has a direct relationship with the process of Requirements Engineering (RE), Risks Management (RM) and Evolution Management (EM) of the RiPLE.

![Figure 5.2 Relationship of the SC with RM, EM and RE](image)

The RE activities are responsible for refining the scope, thus, the scope definition is a pre-requisite to start the RiPLE-RE which has the Product Map
artifact (a matrix of features and products generated by the RiPLE-SC) as a mandatory input.

The RM activities are related to the identification and management of the risks related to scoping as well as all the risks identified during the RiPLE execution. If some risk is identified, it is catalogued by the RM and techniques of management are applied to mitigate it.

The communication between SC and EM is performed by Change Requests (CR) and maintenance tasks in the artifacts created during the scoping phase. If some artifact of the SC is incomplete or inconsistent and this is discovered in some of the phases of the scoping process, the SC requests a change to EM. The EM analyzes the change and when approved, sends a task to SC. Change Requests related to scope can also occur in other disciplines of the RiPLE lifecycle, generating new tasks to the SC.

In the composition of the RiPLE-SC, there are phases which are related to tasks, roles and work products (inputs and outputs), as can be seen in Table 5.1. Moreover, all the phases of the process are iterative, incremental and clearly integrated with agile aspects, such as: review meetings to obtain feedback of customers; creation of user stories to obtain the real needs of the customers; pre-scoping meeting to gather the different visions of the stakeholders; and so on.
### Table 5.3 Summary of the RiPLE-SC

<table>
<thead>
<tr>
<th>Phases</th>
<th>Tasks</th>
<th>Inputs</th>
<th>Outputs</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pre-Scoping</strong></td>
<td>Pre-Scoping Meeting</td>
<td>Domains List, Stakeholders’ Information</td>
<td>SPL Vision</td>
<td>Customer, Scoping Expert, Domain Expert, Architect, Domain Expert, Developer, Product Line Manager, Market Expert</td>
</tr>
<tr>
<td></td>
<td>Analyze Market</td>
<td>Domains List, SPL Vision</td>
<td>Business Plan</td>
<td>Market Expert</td>
</tr>
<tr>
<td><strong>Product Scoping</strong></td>
<td>Construct User Stories</td>
<td>Products List, Customers’ Needs</td>
<td>User Stories</td>
<td>Customer, Market Expert, Domain Expert</td>
</tr>
<tr>
<td></td>
<td>Identify Features</td>
<td>User Stories, Stakeholders’ Knowledge</td>
<td>Features List</td>
<td>Scoping Expert, Domain Expert, Domain Analyst, Architect, Developer, Product Line Manager, Market Expert</td>
</tr>
<tr>
<td></td>
<td>Feature Review Meeting</td>
<td>Features List</td>
<td>Features List</td>
<td>Customer, Market Expert, Domain Expert</td>
</tr>
<tr>
<td></td>
<td>Construct Product Map</td>
<td>Features List, Products List</td>
<td>Product Map</td>
<td>Scoping Expert</td>
</tr>
<tr>
<td></td>
<td>Validate Product Map</td>
<td>Product Map</td>
<td>Product Map</td>
<td>Customer, Market Expert, Domain Expert</td>
</tr>
<tr>
<td><strong>Assets Scoping</strong></td>
<td>Create Benefit and Characterization Metrics</td>
<td>SPL Vision</td>
<td>Metrics</td>
<td>Scoping Expert</td>
</tr>
<tr>
<td></td>
<td>Apply Metrics</td>
<td>Metrics, Product Map</td>
<td>Quantified Product Map</td>
<td>Customer, Scoping Expert, Domain Expert, Architect, Developer, Product Line Manager, Market Expert</td>
</tr>
<tr>
<td></td>
<td>Prioritize Product Map</td>
<td>Quantified Product Map</td>
<td>Quantified Product Map</td>
<td>Customer, Product Line Manager, Scoping Expert, Domain Expert</td>
</tr>
</tbody>
</table>

#### 5.3.1. Phases

RiPLE-SC consists of four main phases that are performed in an iterative way to enable frequent feedback: *Pre-Scoping*, *Domain Scoping*, *Product Scoping* and *Assets Scoping*. Besides, we believe that during the achievement of these phases is very important the use of *reflection meetings*, meetings that should be performed at regular intervals in order to discuss, what is good, what should be
improved and what should be performed to improve, adjusting the behavior of the team adequately. Figure 5.3 shows the phases of the RiPLE-SC.

The pre-scoping aims to identify relevant characteristics which will influence in the next stages of the scoping process, such as business goal, profile of the team, operational and organizational contexts and market aspects. Moreover, this phase proposes to promote an initial contact between the project team and customers.

The domain scoping has as objective to analyze and discuss systematically the domains among the members of the project, considering points pre-determined which aim to help in the definition of the most relevant domains and disseminate the knowledge on the domains with the team, factor which decreases the need of documentation.

The product scoping aims to identify the particular products that will be developed and the features which they should provide. It is oriented with the needs of the customers of the product line and market aspects.

The assets scoping aims to identify the assets that should be developed in a reusable way. In this phase, the scope is aligned with the business goal previously determined by the stakeholders in the phase of pre-scoping.

![Figure 5.3 Phases of the RiPLE-SC](image-url)
5.3.2. Inputs and outputs

RIPE-SC receives a domain list previously formulated as input which, in general, is defined by the domain expert and/or market expert. The domain list determines the domains for further analysis in the product line. Thus, it provides the basis for the scope definition. Based on the domains with better market potential, products and features are identified for a product line. A domain is a set of current and future applications which share a set of common capabilities and data (Kang et al., 1990).

The output of the RIPE-SC is a quantified product map. It presents the final set of products and features of the product line. This product map is also composed of values associated to the metrics applied for features in each product (See Section 5.4.4).

5.3.3. Roles

A role defines the behavior and responsibilities of an individual or a group of individuals working together as a team. The behavior is expressed in terms of tasks which the roles perform, and each role is associated with a set of cohesive tasks [Kruchten, 1998]. During the phases of the RIPE-SC, some roles are essentials as follows:

- **Scoping expert**: The *scoping experts* are responsible to conduct the scoping process, driving the workshops performed in the process. It is important that the scoping expert knows agile techniques and practices, since they will aid in the execution and management of the tasks flow, in the impediments solution and in the interaction among stakeholders;
- **Customer**: The *customers* have critical role in the process and their presence is essential for that the products of the product line present their real needs;
- **Domain Expert**: The *domain experts* provide their knowledge on the domains and the products related to it;
- **Market Expert**: The *market experts* provide knowledge on market analysis and can help in the identification of the domains and products more relevant for a determined market segment;
- **Developer**: The developers are very important for the phase of *assets scoping*, where they define estimative related to effort. Moreover, the
developers can identify technical constraints for features implementation;

- **Architect**: The architects are responsible to determinate which features will constitute the reference architecture;

- **Product Line Manager**: The product line manager is responsible by providing the organizational objectives.

### 5.4. The RiPLE-SC Process

In this section, the phases, tasks, inputs, outputs and roles of the RiPLE-SC are presented in more details.

#### 5.4.1. Pre-Scoping

In this phase, relevant characteristics which will influence the next stages of the scoping process such as project vision, operational and organizational context, stakeholders and roles, business goal and market potential are identified.

The pre-scoping phase is composed of two tasks: **pre-scoping meeting** and **analyze market**. Figure 5.4 shows the inputs, outputs, tasks and roles of the pre-scoping phase. The next sections present the tasks of this phase in details.

![Figure 5.4 Pre-scoping phase](image)

#### 5.4.1.1. Pre-Scoping Meeting

This task has as objective to identify general information related to: 1. **customer**, such as their expectations about the project; 2. **team**, such as the role that performs in the organization and its experience, the knowledge level on the domain, the knowledge level on the process as well as on the techniques that
will be used and their expectations about the project; and 3. organization, such as the current process used in the projects, business constraints and organizational structure and maturity. Moreover, this task aims to provide an initial contact between the customer and the project team and exposes the agenda which will drive the application of the process.

The identification of the information necessary from the customer, project team and organization can be obtained with the use of questionnaires and semi-structured interview (Seaman, 1999) that will drive the pre-scoping meeting. The meeting is conducted by the scoping expert through informal conversation with the stakeholders, factor that encourages the collaboration of the customer in the project and the collaboration inside the team. The active involvement of the customer and the cooperation among all the stakeholders is essential for the success of agile projects (Hunt, 2006). Besides, the communication face-to-face is considered by the Agile Manifesto as the most efficient and effective method of conveying information in a project team.

Four steps should be performed in the pre-scoping meeting: identify organizational context, identify operational context, analyze stakeholders and identify business goals. For performing these steps, the pre-scoping meeting receives as inputs the stakeholders’ information, which are related to information such as “what” are the expectations of the stakeholders for the SPL, “what is the profile of the project team” and in “which” operational and organizational context the project will be inserted; and a domains list, where the domains name that will be analyzed in the product line are listed. The domains have direct influence on the decisions which are performed during the pre-scoping meeting, for example, in the choice of the scoping team, where stakeholders with more experience on the domains will execute the scoping process, since the understanding of the problem is essential for the agility of the process. After the execution of the steps determined in the pre-scoping meeting is produced as output, the SPL vision. The SPL vision represents the general understanding of the stakeholders about relevant information for the project, such as structure of the organization, maturity of the stakeholders on the process that will be used, as well as on the domains, business constraints, business goals and so on.
Identify Organizational Context

It must capture context aspects of the organization for identifying how the activities performed during the identification of the scope will be influenced.

In the analysis of the organizational context, the following aspects should be considered:

- **Structure**: The structure is an aspect that can influence the planning of meetings and workshops. In the context of small organizations, the execution of a more agile approach, i.e., focused on the use of workshops integrated by all the stakeholders of the project, makes possible the communication face-to-face, the collaboration among the stakeholders and, consequently, the decrease of documentation. In complex organizations, dates are harder to find and important experts might miss due to other obligations and sometimes it is not possible that all people meet at the same location (John et al., 2006). Thus, for mitigating these problems, we recommended the utilization of different means, such as workshops and individual interviews for communication and coordination.

- **Maturity**: It directly influences the problem understanding and, consequently, the agility in which the scoping process is performed. The knowledge of the process, as well as, of the domain analyzed is fundamental for the best scoping result. Therefore, the maturity will determinate the need for trainings on the process and the investigation of documentation about the domains before performing the scoping process.

Identify Operational Context

The operational context aspects have impact on the planning of scoping, influencing the overall scoping process (John et al., 2006). They drive decisions that affect all further scoping tasks. These decisions are exposed in the following aspects:

- **Business Constraints**: Business constraints (e.g. time-to-market and resources) can influence the scoping process. They can affect the process lifecycle and the detail level of some assets. Furthermore, it
influences on the choice of stakeholders, tools and in the scope size. In projects where the time and the resources are limited, the team should commit oneself with a smaller scope.

- **Process:** The identification of the process used by the organization determines if there is a relationship between it and reuse practices. In cases where the organization presents reuse practices inserted in the current process, the adoption of SPL practices is easier, as well as the understanding of why specifics tasks of scoping are performed.

**Analyze Stakeholders**

A stakeholder is someone who has a defined interest in the outcome of the project. In the RiPLE-SC, the choice of the stakeholders and their respective roles is performed according to the profile analysis of each stakeholder in the initial context of the product line.

A product line can start from the scratch, i.e., it can be introduced while some products are already under development, or its core assets can be reengineered from legacy systems (Bayer, 1999). Thus, the initial context influences on the identification of the stakeholders and, consequently, in the scope definition, because it can present different information sources. In this choice is also considered the maturity of the stakeholders on the domains which will be analyzed in the SPL.

**Identify Business Goals**

In this step, the goals of the stakeholders of the RiPLE-SC (See Section 5.3.3) are considered.

RiPLE-SC, as well as the work defined by Rommes (2003), consider that scoping is not only an economical activity, but also a social activity and, therefore, should be defined with the participation of each relevant stakeholders group for a SPL, making possible the identification of different views and business goals. In this sense, there are an arbitrary number of business goals for product line development. However, according to the study on software reuse measurement, based on experts, performed by the RiSE Labs (Almeida, 2009), the business goals most considered in organization that adopts practices of product lines are, respectively: **reduce costs**, **improve the productivity**, **improve**
the time-to-market, improve the quality, increase the company portfolio and gain new markets.

5.4.1.2. Analyze Market

The systematic research and the analysis of external factors such as market segments and potential determine the success of a domain in the market. It involves the gathering of business intelligence, competitive studies and assessments, market segmentation, customer plans, and the integration of this information into a cohesive business strategy and plan (Clements and Northrop, 2001). Thus, the task analyze market aims to obtain information of the market segments in which the domains are inserted for identifying issues that can determine their success in the marketplace.

This task uses as input the domains list that will be analyzed in the product line and the SPL vision, producing as output the business plan. The business plan is a strategic plan that has information about the domains potential in the market, the market segments related, their sales channel and the factors that determine the success of these domains in the marketplace.

Market analysis is not a trivial task and is influenced by costs, complexity and maturity of the organization on the domain. However, in a product line, scoping has its base on the areas of domain-specific reuse, economic modeling in reuse, marketing science and, mainly, in product line engineering (John et al., 2006). So, in general, scoping requires market knowledge and, in this context, market experts' knowledge is very important. They frequently interact with users and customers and are essential to represent the business concerns in the product lines adoption. Their understanding of sales records and market trends allow them forecasting future sales and assessing expected benefits.

5.4.2. Domain Scoping

A key question in developing a domain scoping process is: which set of factors can determine the domains and sub-domains with more potential? The answer for this question is to consider dimensions such as: maturity, volatility, market potential, reuse potential, risks, experience, coupling and existing code potential. These dimensions directly impact on the potential of a domain or sub-domain, for example, matures domains make possible easy access to
documentation and the existence of reusable code is more probable. In this sense, the domain scoping phase of the RiPLE-SC combines the fundamental aspects for a domain analysis of success. Some of these relevant aspects are used by others work about scoping such as Schmid (2002) and John et al. (2006).

The domain scoping of RiPLE-SC aims systematically to analyze and discuss the domains and sub-domains among the project members, considering points which aid in the definition of the most relevant ones. Moreover, this phase proposes the dissemination of knowledge on such domain and sub-domain among the team, decreasing the need for documentation.

This phase is composed of the task *analyze domains*, which has as inputs: an already prepared list of domains, the *SPL vision* identified in the task of *pre-scoping meeting* (See Section 5.4.1.1), the *stakeholders’ knowledge*, relevant information of each stakeholder about the dimensions discussed, and the *business plan*, defined in the task *analyze market* (See Section 5.4.1.2). It produces as output the *domains and sub-domains list*, composed of the domains and sub-domains with more potential for the SPL. Figure 5.5 shows this phase.

![Figure 5.5 Domain scoping phase](image-url)
5.4.2.1. Analyze Domains

This task, as well as occur in other process of scoping (Schmid, 2002; John et al., 2006), is based on the domain analysis performed in processes of Domain Engineering such as Neighbors (1980), Kang et al. (1990) and Almeida (2007).

The term domain analysis was first introduced by Neighbors (1980) as “the activity for identifying the objects and operations of a class of similar systems in a particular problem domain”. A domain is a set of current and future applications which share a set of common capabilities and data (Kang et al., 1990).

In the RiPLEySC, the task analyze domains as well as the other tasks of the process is iterative and incremental. Therefore, in each iteration new information can be searched and thus evidenced the need of inclusion or exclusion of some domain. For example, it is common that in the definition of the products, new domains are identified and thus an analysis of these domains should be performed. It is possible also that with the prioritization of the features, a domain is considered dispensable and excluded.

The domain analysis of the RiPLEySC uses the workshops culture in its execution, i.e., all steps of the task analyze domains are performed in the workshops format, mediated by the scoping expert. The workshops are a fundamental resource from agile methodologies for maintaining the integration in a team. Besides, it allows that the stakeholders of the project discuss several aspects expressing their viewpoints. Thus, it is important the presence of each representative stakeholder of the process in this task (See Section 5.3.3). The workshop culture is also used in methods such as: DSDM, where it is considered together with prioritization, timeboxing and prototyping, the heart of the method (Norfolk, 2001); and in the Crystal methodology, where is used to discuss the work of the team and perform reflections on the project (Cockburn, 2004).

The domain analysis workshops proposed in the RiPLEySC makes possible the communication face-to-face and the collaborative work between business and technical people. It has as goal to identify the domains and sub-domains more relevant for the product line. In addition, this analysis requires a clear understanding of the stakeholders that integrate the workshops of the domains, because the agility can be negatively influenced when the stakeholders
do not have experience with them. Thus, before the beginning of the domain analysis, the participants should explore information on the domains which will be analyzed. This information can be obtained from: project plans, user manuals, modeling, data dictionary, existing applications and knowledge from domain experts.

Four steps should be performed in the task *analyze domains: review domains, identify sub-domains, analyze sub-domains* and *prioritize domains and sub-domains*. These steps are discussed in the next sections.

**Review Domains**

In this step, the objective is to discuss general characteristics of the domains, making possible to align the knowledge of the stakeholders regarding them.

The review of each domain is performed in pre-determined timeboxes. With them is possible to limit the time of the analysis and maintain the focus on the workshop.

In the review, the stakeholders should express their understanding on the domains, enabling them to identify new domains, exclude existent ones or maintain the initial set of domains.

**Identify Sub-Domains**

A domain can be composed of a big number of systems. Furthermore, the identification of “what” is in and out of a domain depends on the vision of specialists and stakeholders in general, because each one has their particular interests (Lucredio, 2009). Thus, the division of the domains in technical sub-areas (sub-domains) makes possible a deepened analysis of each individual area of the domains, facilitating the choice of the areas where the reuse is adequate and where the economical potential is most suitable. In this sense, the process helps to reduce the effort required in performing their next phases, because sub-domains can be previously excluded of the product line according to the potential of them.

The sub-domains definition is performed in brainstorm sessions. These brainstorms provide an opportunity for the stakeholders, together, gather information about their needs and expectations regarding to key sub-domains that are of particular concern to them. In this step, is important also that the
team is aligned with the needs of the customers and the product line as whole. Thus, the knowledge of domain experts and market experts is determinant to drive the team in the choice of which sub-domains are more relevant for SPL and their customers.

**Analyze Sub-Domains**

In this step, each sub-domain is individually discussed concerning the relevant dimensions previously defined for the analysis. For the discussion, the scoping expert defines a time interval (timebox) that considers sufficient for the analysis of the sub-domain in question. In this time interval, the sub-domain is discussed according to each dimension by the stakeholders that integrate the domains analysis workshop. Thus, considering the different visions of the stakeholders, a conclusion about each dimension evaluated in each sub-domain is performed. The following dimensions are considered in the analysis:

- **Experience:** It indicates the level of knowledge that the participants of the workshop have on the sub-domain. This level can be:
  1. *High*, the participants have fully understanding and some stakeholders participated of projects related to the sub-domain;
  2. *Partial*, the sub-domain is knowledge, but the understanding is partial; or
  3. *Low*, the participants do not know the sub-domain.

- **Risks:** They are identified and analyzed to determine the negative impact on the sub-domain. In the analysis, the risks are prioritized according to the perceptions of the team about their severity. These impacts can be:
  1. *High*, possible problems will happen and will be difficult to manage;
  2. *Relevant*, possible problems can be expected, however, they can be managed; or
  3. *Low*, there are not apparent problems.

- **Volatility:** It determines if the sub-domain changes with the time. The volatility can be:
  1. *High*, constant changes occur in the sub-domain;
2. *Medium*, possible changes can be expected, but, they do not have big impacts on it; or
3. *Low*, few changes can occur.

- **Maturity**: It determines if the sub-domain is stable. The maturity can be:
  1. *High*, the participants have fully understanding, materials about the domain are of easy access and some stakeholders participate in projects related to the sub-domain;
  2. *Partial*, the sub-domain is known, but the understanding is partial and the access for information about it is more difficult; or
  3. *Low*, the participants do not know the sub-domain.

- **Code Potential**: The code existence facilitates the understanding of the sub-domain and aids in the development. The code potential can be:
  1. *High*, is an interesting basis of information and has potential to be reused;
  2. *Relevant*, some costs are associated to make it reusable; or
  3. *Low*, the code does not aggregate benefits to the project.

- **Market Potential**: It identifies which sub-domains can obtain greater economical return in the market segments. The market potential can be:
  1. *High*, it has strong acceptation in the marketplace and several products can be expected;
  2. *Medium*, there is some market potential, but, few systems can be expected; or
  3. *Low*, there is few or no clear market potential.

- **Reuse Potential**: The reuse potential determines the possibility of the sub-domains of composing a generic reuse infrastructure. The reuse potential can be:
  1. *High*, a generic reuse infrastructure is possible and the effort for making it is low;
  2. *Medium*, a generic implementation is possible, but only for some sub-areas; or
  3. *Low*, it will be probably impossible a reuse investment.

- **Coupling**: The coupling identifies if the sub-domains can be dealt in an independent way. The coupling level can be:
1. High, it can only be defined related to others;
2. Medium, there is a considerable low coupling, but it can be related to other sub-domain;
3. Low, the sub-domains are dealt in an independent way.

The dimensions considered in the evaluation of the sub-domains were defined with base on relevant aspects identified in the analysis of the literature systematic review (Chapter 3). With these dimensions is possible to obtain a clear evaluation of the sub-domains with more potential for the product line. Besides, with the workshop format and the use of timeboxes is possible to discuss high amount of information in few time, conferring agility for this step.

**Prioritize Sub-Domains**

The prioritization is performed based on the results obtained in the sub-domains analysis, the business goals and considerations made by the customers in the task of pre-scoping meeting (See Section 5.4.1.1).

This step is performed by different stakeholders and is possible that the different viewpoints result in conflicts. Thus, it is recommendable an initial search for common interests and the negotiation of the divergent ones. In this case, discussion sessions moderated by the scoping expert can be made, enabling the alignment of the interests for all the team. With these sessions is constructed the final list of domains and sub-domains more relevant for the product line.

**5.4.3 Product Scoping**

In today’s competitive business environment, it is extremely important to offer for customers exactly the products that they want. SPL has the potential to enable companies to offer a large variety of products while still being able to manage the complexity caused by this increased number of products. In this sense, the product scoping phase aims to define the product portfolio that optimally satisfies customer demands, characteristic of highest priority in agile process, where the customers’ role is critical (Nerur et al., 2005) and, at the same time, restricts the number of products offered.

In this phase, the products and features associated with them are identified and evaluated with respect to their potential for introduction in the
product line. In this identification and evaluation, the market analysis as well as the real needs of the customers are considered. The choice for using features as the form to represent products has several reasons:

- Features provide a simple language, of easy understanding and communication for the stakeholders;
- Product portfolios are generally defined in terms of features; and
- Features provide a well established approach for refinement and description of requirements.

In order to define the product scoping, five tasks should be performed: construct user stories, identify features, features review meeting, identify products, construct product map and validate product map. Figure 5.6 shows the inputs, outputs, tasks and roles of the product scoping phase. The next sections present the tasks of this phase in details.

5.4.3.1. Construct User Stories

In a product line started from scratch or in its evolution, the utilization of user stories as basis for identification of customers’ needs and, consequently, features of product is very important. It is visible the fact that customers have big difficulties of expressing their expectations before using the final product (Nuseibeh and Easterbrook, 2000). Thus, in these cases, user stories should be used, because in our vision, this is the most natural way for customers express their needs. However, in cases where it is necessary to construct a product line of pre-existent products, i.e., in cases of reengineering, the construction of user stories is unusual, and the features are identified from the analysis of documents associated to products or through the code.
RiPLE-SC defines user stories as brief descriptions from the customers of how they will use the system (Hunt, 2006), as used in eXtreme Programming (XP). The stories present short names, business value and are written in a short prose or description graphical. As can be seen in Figure 5.7, the stories are not estimated for effort as occur in XP. Effort as well as other business goals are measured in this process in the phase of assets scoping across a set of metrics based on GQM (See Section 5.4.4).

![Figure 5.7 User story (Beck and Andres, 2004)](image)

The identification of business value, related to the stories, is a tentative of focusing on the importance level that they have for the customer. With this information is possible filter which issues are really indispensable for the customers and decrease the implementation and planning effort across the exclusion of stories that can be omitted without affecting the products derivation. The business values associated to the stories are:

- **High**: This story is absolutely needed for the customer. He will not enable the derivation of an useful product without this story.
- **Medium**: This story is important for the customer. Including this story into the final system will provide verifiable benefits to the customer.
- **Low**: This story clearly provides additional value to the customer, but it can be omitted without affecting the main uses of the derived products.

Each story can be written in index cards because they are relatively small, easy to move and order (Beck, 1999). If there is the need to report the progress to others parts of the organization in a traditional format, electronic media can be used.
5.4.3.2. Identify Features

The goal of this task is to determine the features which will be present in the further products of the SPL.

The identification of the features is performed across a workshop whose collaborative participation of the team is essential. This workshop should be moderated by the scoping expert, and all participants are expected to be fully engaged and present throughout the features identification step. Participants are encouraged to comment and ask questions at any time during the workshop. However, it is important to recognize that the moderator may occasionally have to cut discussions in order to control the time or when it is clear that the discussion is not focused on the required workshop outcomes. Moderator should also help stakeholders to create well-formed features.

During the workshop is possible to identify and discuss several issues with base on different perspectives. In this sense, different views of the stakeholders will be confronted and analyzed enabling a better scoping result.

For identifying the features, RiPLE-SC uses as base: 1. user stories, in this case an evaluation of the user stories by representative stakeholders is essential to determine if the stories are feasible and complete, and thus to identify potential features for the product line and create competitive products; and/or 2. the abstraction of the previous knowledge obtained from documents such as books, user manuals, design documents and code (Lee et al., 2002). In this context, the Analysis Guidelines (AG) defined initially by Lee et al. (2002) can be very useful:

**AG1.** Analyze terminologies used in the domain to identify features. In some mature domains, experts normally use standardized terminology to communicate their ideas, needs, and problems. Thus, using standard terms for the feature identification can expedite communication between the project team and information providers (domain experts, end-users and customers).

**AG2.** Try to first find differences among existing applications in a domain and then, with this understanding, identify commonalities. Applications in the same domain share a high level of commonality (Coplin et al., 1998). Hence, the commonality space would be larger to work with than the difference space, thus finding difference is much easier than finding commonalities. So, the strategy is, initially, to identify the existing applications (running systems,
documentation, etc), and then list different features that characterize each one. After this understanding, the identification of common features is easier to be performed.

**AG3.** Do not identify all implementation details that do not distinguish among applications in a domain. A developer tends to enumerate all the implementation details and to identify them as features, even though there are no variations among them. But it is important to note that a feature model is not as detailed as a requirement model, where desired functionality is expressed for developers to understand and implement it.

In this process, functional and non-functional features are considered. The functional features are related to aspects as “what” the system has to do. The non-functional features are associated with quality attributes, which the products should address. These attributes will serve as architectural drivers of the product line and present impact on the product line architecture. Therefore, the identification of non-functional features is of extreme relevance for the architecture of the product line. Some types of quality attributes are presented in Table 5.2.

<table>
<thead>
<tr>
<th>Quality Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>It is related to the facility that users have of using the system.</td>
</tr>
<tr>
<td>Performance</td>
<td>It is related to the response time of the system.</td>
</tr>
<tr>
<td>Security</td>
<td>It is the ability of the system to prevent or resist unauthorized access while providing access to legitimate users. An attack is an attempt to breach security.</td>
</tr>
<tr>
<td>Portability</td>
<td>It is related to the ease of porting the software to other host machines and/or operating systems (IEEE Std. 830, 1998).</td>
</tr>
<tr>
<td>Availability</td>
<td>It is related to the amount of faults of the system and its duration.</td>
</tr>
<tr>
<td>Maintainability</td>
<td>This should specify attributes of software that relate to the ease of maintenance of the software itself (IEEE Std. 830, 1998).</td>
</tr>
</tbody>
</table>

It is important to highlight that features are not requirements and should be identified and described in high-level details. Thus, implementation details should be not listed as features. This is the reason because the workshop for
features identification is conducted by the scoping expert that drive the team, inhibiting this and other details types.

The feature list identified in this task presents the name of the feature, its description, and priority. Generally, the feature description is performed in situations of immaturity of the team in the domain or in cases where the feature name can cause ambiguity regarding the functionality associated to it. RiPLE-SC defines the features description as optional, because this step requires high effort and could affect the agility essence of the process. As can be seen in Table 5.3, the feature “delete submission” does not present description, because its functionality in the system is auto-explained by its name. The feature priority presents the importance level for the product line and customers, and can be defined as: high, medium or low as occur with the user stories (See Section 5.4.3.1). Table 5.3 shows an example of feature list for the conference management domain.

Table 5.5 Features list

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>List events opened for submission</td>
<td>It lists all events opened for submissions</td>
<td>High</td>
</tr>
<tr>
<td>Create event</td>
<td>The system enables to create new events</td>
<td>High</td>
</tr>
<tr>
<td>Create event based on a previous</td>
<td>The system enables to create events with the data base of previous event.</td>
<td>Medium</td>
</tr>
<tr>
<td>Profile</td>
<td>For each event, the user can have a different profile (e.g. author, PC Member, Admin, etc).</td>
<td>High</td>
</tr>
<tr>
<td>Authors management</td>
<td>Allows add/remove authors and edit information about him.</td>
<td>High</td>
</tr>
<tr>
<td>Author indicates interest conflict</td>
<td>The authors can indicate reviewers who they have conflict of interest during the submission step.</td>
<td>Low</td>
</tr>
<tr>
<td>Attach files</td>
<td>It allows the submitter to add the article files.</td>
<td>High</td>
</tr>
<tr>
<td>Submitted papers</td>
<td>A list with the submitted papers (by an author) is provided.</td>
<td>Medium</td>
</tr>
<tr>
<td>View submission</td>
<td>It opens a file with the submitted paper and additional information.</td>
<td>High</td>
</tr>
<tr>
<td>Delete submission</td>
<td></td>
<td>High</td>
</tr>
</tbody>
</table>

5.4.3.3. Feature Review Meeting

The feature review meeting is related to feature evaluation and negotiation between the organization (domain expert and market expert) and customers.
Domain experts and market experts are indicated for this task because they present high knowledge of the domains, potential market segments and their needs. Their participation aids in the identification of a correct scoping which maximizes the economical return of the SPL focusing on the customer’s need.

The main goal of this task is to obtain the agreement from the customer about the feature list.

Constant feedback is an agile principle which enables customers to evaluate the work performed and define if the releases are aligned to their real needs. With the review meeting, new features can be included, excluded or reprioritized in the list.

5.4.3.4. Identify Products

The task *identify products* has as objective to find appropriated products for the product line. This task receives as inputs the market experts’ knowledge, the SPL vision, the features list and the business plan. The output of this task is a product list that presents the products of more potential for the SPL.

The choice of products for a products line is critical for the organizations, because the market is competitive and requires products diversity, factor that can have negative consequences, such as: increase of costs, complexity and time-to-market, causing decrease of benefits with the product line (Jiao et al., 2005). Thus, in this choice is highly important to consider a set of products aligned with the goals of the organization and needs of a specific group of customers or market segment. In addition, the products can be identified according to the following scenarios:

1. The organization has similar products, but they are developed as unique products. In this scenario, the products can be based on the available ones;

2. The product line is started from scratch. In this scenario, the products can be defined across customer segments based on the features determined by them, i.e., customers with similar needs are grouped in a customer segment and from each segment is derived one product.
5.4.3.5. Construct Product Map

After the identification of the products and their features, these are organized in a product map, used to represent which features will integrate or not each product. In this map, columns and rows are used to represent products and features, respectively. Moreover, in this map, each column is constituted of two other columns, the first indicates if the feature is one possible future feature of the product and the second indicates if the feature is required in the product, i.e., if the feature is required in a product “X”, the row representing it will present the number “1” in the column “Req” and the number “0” in the column “Fut” of the product “X”, in cases where the feature is a future feature, the column “Fut” receives “1” and the column “Req” receives “0”. Still there are cases, where the feature is not part of a determined product, in these cases, both the columns “Fut” and “Req” receive “0”.

In addition, in the product map, each feature is associated at a scope as follows:

1. **Mandatory**: Features that are required by all members of the product line;
2. **Optional**: Features that are part of some products, i.e., can or cannot be selected by the products of the SPL; or
3. **Out of Scoping**: Features that are part of only one product.

As can be seen in Figure 5.8, the product map is a consolidated view of the results obtained during the phase of product scoping.

<table>
<thead>
<tr>
<th>Features</th>
<th>RSE Chair Conference</th>
<th>RSE Chair Journal</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fut</td>
<td>Req</td>
<td>Fut</td>
</tr>
<tr>
<td>Access Control</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Accept/Reject Review</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Assignment - Automatic Indication</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Assignment - Chair Indication</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Assignment - Preference Indication</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Best Papers</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Build PDF</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Comments to Author</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Complete Submission</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Create Event from Previous</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Create Event From Scratch</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Deadline</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Delete Submission</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Document Similarity Analysis</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Event Action History</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Event Opened for Submission</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 5.8 Product map**
5.4.3.6. Validate Product Map

The product map validation is performed in meetings with the participation of customers, domain experts and market experts. These meetings have as objectives the following aspects: 1. identify if any feature was forgotten or allocated improperly in some product; and 2. verify the scope defined for the features (mandatory, optional or out of scoping).

With this task, the product map is consolidated enabling the choice of the assets that will integrate the reuse infrastructure for the SPL.

5.4.4. Assets Scoping

The goal of the assets scoping is to determine the appropriate features that should be built for reuse. Thus, the assets scoping establishes the reusability of features relevant for the development of the reference architecture, maximizing the economical return obtained with the product line.

This phase is based on a quantitative analysis of the benefits of making the feature reusable. Therefore, the combination of features that provide the best benefits constitute the assets scope. In this sense, our process determines the definition of metrics for measuring the benefits that specific feature has for the product line. It aims to optimize the product line according to specific benefits.

It is visible that projects involve different visions from different stakeholders and that benefits can differ according to business goals of them. In this context, we consider GQM as the solution for deriving metrics based on different business goals expected by the stakeholders.

The choice of performing assets scoping focused on GQM was influenced by the study performed in (Almeida, 2009), where ten experts were surveyed regarding some factors related to software reuse measurement and was identified that GQM is the most utilized method for reuse measurement, as can be seen in Figure 5.9.
This phase is composed of the tasks: create metrics, apply metrics and prioritize product map. The inputs, outputs and roles of these tasks can be seen in Figure 5.10.

**Figure 5.10 Assets scoping phase**

### 5.4.4.1. Create Metrics

The metrics creation involves two well-defined steps: refine and operationalize business goals.

**Refine Business Goals**

The task create metrics initiates with the step refine business goals. In this step, the goals which were previously produced in the phase of pre-scoping are analyzed and refined. The refinement is made because in the phase of pre-scoping the goals are identified generically. Therefore, in this step, the goals are...
refined according to their relevance for the customers, organization and domains in the particular context.

**Operationalizate Business Goals**

After the refinement, it is performed the operationalization of the goals. In our process, it is based on GQM (Basili et al., 1994) and PuLSE-ECO (Schimd, 2002a) following four distinct levels:

1. Goal, where the goals are described in more details;
2. Question, used for characterizing and refining the goal;
3. Characterization metrics, which are elementary metrics where the values can be captured in the environment. They are used to determine the values for the benefit metrics; and
4. Benefit metrics, which describe in a formal way the measures to be used to determine the goals achievement;

In this context, the operationalization flow starts with the goal level, where the optimized description of the goal is performed following the schema that has the form of <Purpose><Issue><Object><Context>, such as *Minimize the effort needed for the development of new applications from the viewpoint of software engineers in the company.*

On the question level, additional aspects are elicited and the goal is defined more preciously. After the aspects have been elicited, the goal is expressed in characterization metrics that will be used in the benefits metrics.

The characterization metrics are used for identifying the need of a specific feature to integrate a product of the product line. The benefit metrics describe the potential of introducing a specific feature into reuse infrastructure considering a defined business goal.

Table 5.4 shows one example of this step in the repository domain. The template developed for this step can be found in Appendix C.
### Table 5.6 Benefit and characterization metrics (Almeida, 2007)

<table>
<thead>
<tr>
<th>Objective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To attend most of the domain applications, maximizing the repository use and standardizing the characteristics that are available in the applications</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. How to define if the repository support or not an application?</td>
</tr>
<tr>
<td>2. How to evaluate the level of use from the repository by the applications?</td>
</tr>
<tr>
<td>3. How to determine the deviation of a characteristic of an application in relation to the established pattern?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Characterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Is characteristic c necessary for the application a?</td>
</tr>
<tr>
<td>- req(c, a) = 1: yes; 0:no</td>
</tr>
<tr>
<td>• Similarity of the characteristic cᵢ with the standard characteristic c in the application a</td>
</tr>
<tr>
<td>- sim(c, cᵢ, a) = 1: equal the standard; 0.5: next to the standard; 0:different of the standard</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Proximity of the characteristic in the applications in relation with the standard</td>
</tr>
<tr>
<td>- P(c, a) = ( \sum_{a \in A} \text{req}(c, a) \times \sum_{c' \in C} \text{sim}(c, c', a) )</td>
</tr>
<tr>
<td>• Distance of the characteristic in the applications in relation with the standard</td>
</tr>
<tr>
<td>- D(c) = ( \sum_{a \in A} \text{req}(c, a) \times (1 - \text{sim}(c, c', a)) )</td>
</tr>
</tbody>
</table>

#### 5.4.4.2. Apply Metrics

In this task, the benefit and characterization metrics are applied for the product map in two steps: apply characterization metrics and apply benefit metrics.

**Apply Characterization Metrics**

In general, this task is performed by the stakeholders who provided the different goals, e.g., the developer can be useful to estimate effort, while the architect can help in the definition of which are the impacts of some features in the reuse infrastructure.

**Apply Benefit Metrics**

After the application of the characterization metrics, the benefit of the assets for the product line are evaluated. In this step, the metrics are applied based on the goals previously considered.

#### 5.4.4.3. Prioritize Product Map

After assigning the values for the benefit metrics, the task prioritize product map is performed. In this task, the features of more potential for the product line are selected.
The product map prioritization is performed in a meeting which involves the customer, manager, market expert and domain expert, where considerations and negotiations are performed.

As RiPLE-SC has as one of the main goals to satisfy the real needs of the customers, the feedback of them is indispensable for the definition of the scope of the SPL. Thus, the base to perform the prioritization of the product map is the customers’ need.

5.5. The Agility of the Process

We discussed the relevance of adopting an agile process for SPL scoping and showed that SPL and AM can be joined, reinforcing others relevant work in the field as the work of McGregor (2008). It considers that product lines and agile aspects can generate effective hybrid methods that are at the same time agile and asset-based.

In RiPLE-SC, the agility is addressed with the adoption of some agile principles and practices such as:

- One of the main goals of the RiPLE-SC is in the search to satisfy the customers. In this sense, the customer role is indispensable in the process. In order to keep the active participation of the customers during the application and drive the project team to get their real needs, constant delivery of artifacts are performed and consequently constant feedbacks are obtained across meetings for assets validation. Moreover, with the task to construct user stories, the customers can describe their needs for the team and indicate their business values. Thus, we believe that our process supports well the agile principle “highest priority is to satisfy the customer through early...”.

- The principle “welcome changing requirements...” is more related to application engineering. But, in our process, changes can be made in all artifacts generated during the agile scoping. The change management is performed by the RiPLE-EM (See Section 5.3) and the changes can occur during all the RiPLE lifecycle. In fact, changes can be made, for example, in the domains list or / and in the product map while developing core assets in domain
engineering. In this way, our process is conducted in an iterative and incremental way. For instance, some steps are associated to features review and features re-prioritization.

- The constant refinement proposed by this process makes possible the high quality level of scoping, supporting the agile principle “continuous attention to excellence”.

- Across workshops, meetings and interviews, we search the collaborative work between business and technical people and the face-to-face communication. Furthermore, they make possible to align the knowledge of the team for the SPL domains and customers’ needs. Thus, we believe that the principles “...face-to-face conversation...” and “business people and developers must work together daily throughout the project” are well addressed by our process.

- Reflection meetings are used in RiPLE-SC in order to adjust the team’s behavior and improve time and process, satisfying the principle “...at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly”.

- The principle “build project around motivated individuals...” is incorporated in our process when we identify the project team based on the domains which will be analyzed in the product line. Selecting people with high knowledge and experience in the SPL domains is a relevant factor for maintaining the team motivated. In choice of team is also important to consider pro-active people, in order to satisfy the principle “...self organizing teams”.

- The principle of “simplicity” can be identified in the RiPLE-SC with its constant search to decrease effort with documentation and implementation. The artifacts produced in the process are of easy documentation. Moreover, with the use of metrics is possible to decrease the implementation effort of the assets, because only the ones with high potential will integrate the product line.
5.6. Chapter Summary

Defining product line scoping in an agile way is a key aspect for organizations to maximize the benefits with the SPL adoption. With an agile scoping process, the high time and cost related to the product line planning can be decreased, motivating organizations in the adoption of SPL. Moreover, the definition of a systematic process facilitates this adoption by organizations.

In this context, this Chapter describes RiPLE-SC, a SPL agile scoping process. It aims to define systematically scoping based on a set of guidelines, metrics, tasks, roles, inputs and outputs, introducing in these tasks, agile techniques and principles.

Next Chapter will present the case study related to this process.
A process will evolve over time based on feedback and learning from applying it and analyzing the results. In this sense, a case study can be used to analyze it and report its results in environments do not controlled. Thus, case studies are particularly important for industrial evaluation of software engineering process and tools because they can avoid scale-up problems (Kitchenham and Pickard, 1998).

In this context, this chapter presents an industrial case study in order to evaluate the viability of the RiPLE-SC in a software product lines project.

6.1. Introduction

A case study is a suitable research methodology for software engineering research since it studies contemporary phenomena in its natural context (Runeson and Host, 2009).

In order to conduct the case study, we used some of the guidelines proposed by (Kitchenham, 1995; Kitchenham and Pickard, 1998). These guidelines consist of the following steps: (1) “Define the case study context”, (2) “Define the hypotheses”, (3) “Plan the case study”, and, (4) “Analyze and report the results”.

In the next sections, these guidelines are explicated in details.

6.2. Context

In order to define the case study context, it is believed that it should be conducted in accordance with a specific goal and a set of questions that represent the operational definition of the goal. The goal defines what the case
study sponsor wants to know and it is characterized as the basis for the case study hypotheses. In this sense, the GQM method (Basili et al., 1994) was used to define the goal of the case study and derive the questions that should be answered to determine whether the goal was achieved.

### 6.2.1. Goal

A goal is defined for an object, for several reasons, with respect to quality attributes, from various points of view and related to a particular environment. Some objects of measurements are, for instance, projects, methods and technologies (Wohlin et al., 2000). Thus, we defined the goal as follows:

**G.** Analyzing the RiPLE-SC for the purpose of evaluating it with respect to its understandability, effort, completeness and effectiveness from point of view of the researcher in the context of a software product lines project in a small organization.

### 6.2.2. Questions

In order to evaluate the RiPLE-SC process and achieve the goal previously defined, we defined qualitative and quantitative questions. The first ones concerned with the subjects’ feedback about the adoption of the process, and the last ones are related to the data collected during the period that the case study will be performed. These questions try to characterize the objects of measurement with respect to quality factors and determine its quality from the selected point of view (Wohlin, 2000). The questions are presented as follows.

**Understandability**

**Q1.** Do the subjects have difficulties to understand/apply the process?

**Q2.** Are the subjects satisfied using the process?

**Q3.** Did you feel sufficient guidance during the application of the process?

**Effort**

**Q4.** How much effort does it take to apply the process and how it is distributed along the RiPLE-SC process?
Completeness

Q5. Is there some task, role or artifact that would may to be in the process, but was not defined in it?

Effectiveness

Q6. Does the process aid in the execution agility of the tasks, i.e., it is effective in the purpose to be an agile process?

6.2.3. Metrics

Once the questions have been developed, the next step is to associate the questions for appropriate metrics. A set of metrics is associated with every question in order to answer it in a quantitative way, either objectively or subjectively, as stated by (Wohlin et. al, 2000). In this case study, the following metrics were defined:

M1. Process Understanding and Application Difficulties (PUAD)

In order to identify possible misunderstandings in the process use, it is necessary to identify and analyze the difficulties found by subjects applying the process.

\[ \text{PUAD} = \text{Number of subjects with difficulties raised during the process use.} \]

M2. Satisfaction (S)

Satisfaction is the users’ response to interaction with the process. Thus, this metric is proposed to evaluate the subjects’ satisfaction using the process.

\[ \text{S} = \text{Subjects’ satisfaction distribution according to a defined scale: very satisfied, satisfied, unsatisfied, very unsatisfied.} \]

According to (Vegas and Basili, 2005), subject’s satisfaction should not be considered for establishing a hypothesis, because it can be refuted by means of statistical evidence. Thus, this aspect will be assessed informally with the questionnaire use (See Appendix D), examining the opinion of each subject.
M3. **Guidance (G)**
The guidance is related to the training and quality of the process documentation.

\[ G = \text{Subjects' guidance distribution according to the frequency scale (Kitchenham and Pfleeger, 2008): most of the time, sometimes, rarely, never.} \]

As subjects’ guidance can be refuted by means of statistical evidence, this metric will be assessed examining the opinions of each subject. Thus, any hypothesis will not be defined for this aspect.

M4. **Effort to Apply each Phase of the Process (EAPP)**
This metric measures the amount of time spent in order to understand and perform each phase of RiPLE-SC process and produce the artifacts proposed in each specific phase.

\[ \text{EAPP} = \frac{\text{Total Time Spent Applying Phase}}{\text{Total Time Spent Applying RiPLE-SC}} \]

As we do not have any baseline we did not define also any hypothesis for this metric.

M5. **Tasks, Roles and Artifacts Missing (TRAM)**
This metric aims at identifying the tasks, roles and artifacts identified as missing in RiPLE-SC by the subjects. With this metric, we intend to identify every task, role and artifact considered absent from RiPLE-SC in order to include them, depending on the analysis.

\[ \text{TRAM} = \text{Number missing tasks/role/artifact during the application of the process.} \]

M6. **Effectiveness (E)**
This metric aims to identify if the RiPLE-SC process is effective with the purpose to be an agile process, i.e., if it aids in the agility of the tasks.

\[ E = \text{Process effectiveness according to the evaluation scale (Kitchenham and Pfleeger, 2008): excellent, good, inferior, terrible.} \]

As the effectiveness can be refuted by means of statistical evidence, this metric will be assessed examining the opinions of each subject. Thus, any hypothesis will not be defined for this aspect.
6.3. Hypotheses

Defining the hypotheses correctly is the most difficult and important part designing a case study. If the hypotheses are wrong, the case study will not produce useful results (Kitchenham et al., 1995). The definition of the hypotheses must be detailed enough to make clear which measurements are needed to demonstrate the expected effect. The hypotheses can be defined in terms of null hypotheses and alternative hypotheses. The formal case study data analysis and evaluation addresses the null hypotheses, however, it is also important to present the findings in terms of the alternative hypotheses (Kitchenham et al., 1995).

6.3.1. Null Hypotheses

These are the hypotheses that the experimenter wants to reject with as high significance as possible (Wohlin, 2000). Formally, the hypotheses can only be disproved. In this study, the null hypotheses determine that the use of the RiPLE-SC does not produce benefits that justify its use, presenting a poor understandability, effort, completeness and effectiveness. It is important to emphasize that these metrics were never used before and thus there is no baseline for it and an arbitrary value was chosen based on practical experience and common sense. Therefore, according to the case study definition, the following hypotheses can be defined:

\[ H_{01}: \mu_{PUAD} \geq 40\% \]
\[ H_{02}: \mu_{TRAM} \geq 2\% \]

(Having in mind that RiPLE-SC has 11 different tasks, 7 roles and 8 artifacts, we came to the value of 10% as being a reasonable number.)

6.3.2. Alternative Hypotheses

According to Kitchenham et al. (1995), research is proposed and funded based on studying the alternative hypotheses that are stated in favor of which the null hypotheses are rejected (Wohlin, 2000). In this study, the alternative hypothesis determines that the use of the process produces benefits that justify its use. Thus, the following hypotheses can be defined:

\[ H_{11}: \mu_{PUAD} < 40\% \]
\[ H_{12}: \mu_{TRAM} < 2\% \]
6.4. Planning the Case Study

After the definition of the case study, the planning is started. As reported by Kitchenham et al. (Kitchenham et al., 1995), the plan identifies all the issues to be addressed in order to have a smooth evaluation, including the context, training requirements and the data-collection procedures.

6.4.1. Context

According to Kitchenham and Pickard (1998), the case study context sets the goals within which the case study must operate.

In order to define the context to perform the case study, the first step was to select the most suitable company. In this sense, we performed an initial search in a government website which lists more than 100 companies in the information and technology cluster. Based on this analysis, we could identify a set of companies working on products for a specific domain, the right environment to start a product line project.

After that, we tried to contact the top 10 companies in the list and schedule some meetings in order to present a light business case for the project. In this sense, we concentrated the presentation in the product lines ideas, some successful cases from hall of fame (Weiss et al., 2006), and the project idea concentrated in the scoping phase. We believe that scoping is the most important phase to identify the company potential to start a SPL project.

In fact, we defined some meetings with two competitors companies working on the medical domain. The first one is a medium size company and the second one a small. In the first one, the work decision process was too long and the company also passed for a migration step of its products for the web environment, performing some changes in the internal structure. In the second one, an important point was that our main contact and project sponsor is one of the company owners and was the previous technical leader of the products. After his agreement, we had one other meeting with another owner responsible for marketing activities. He also understood the project vision and was negotiated a cooperation project between RiSE Labs and the MedicWare Informatics System LTDA5.

5 www.medicware.com.br
MedicWare works with software development for the medical domain since 1994 and is located at Salvador, Bahia, Brazil. The company was created in 1994 offering strategic and operational solutions integrated for hospitals, clinics, labs and private doctor’s office.

The company has four products presented in the decreasing order according to their size. The first one (SmartHealth) is a product of 35 modules or sub-domains, and manages the whole area of a hospital, from financial to patient aspects. The second one (SmartClin), composed of 28 modules, performs the clinical management supporting activities related to medical exams, diagnostics and so on. The third one (SmartLab) has 28 modules and integrates a set of features to manage labs of clinical pathology. Finally, the last one (SmartDoctor) is the only web product and composed of 11 modules to manage the tasks and routines of a doctor’s office. Figure 6.1 shows the correlation among the products. As it can be seen in the figure, SmartHealth is the biggest product and some of its features compose the other ones.

![Figure 6.1 Correlation among products](image)

The company organizational structure is composed of a partner group, one executive leader, a management group, an operational group, a marketing group, a software development team and a support, training and deployment team.
The company has two main channels to interact with customers. The first one is the business analyst whose primary job is to understand the domain and requests for new features. The second one is a web-based system working as an open channel for all the customers report defects, request new features and product support. The company does not have a systematic software development process, however, the work flow defined for us and validated by them, summarizes the activities as can be seen in Figure 6.2. It starts with a request call for new features or report of a defect in some product. When the request is related to new features, it goes toward for the business analyst and manager. The requests are analyzed and depending on the analysis, the manager makes a detailed specification for new features and sends it to the development and tests group. In fact, the company does not have a formal test team, but the development team is also responsible to perform it. When the request is related to defect in the product or difficulties to use it, the request is sent to the support team which analyzes it and depending on the situation solves the problem for the customer or re-send it to the development and test team.

The company web system has more than 50,000 records with customer requests.

Figure 6.2 Work flow of the MedicWare
6.4.2. Training Requirements

In this case study, we provided the complete description of the RiPLE-SC process for training of the subjects. In addition, all subjects were trained to use the RiPLE-SC by a process expert that worked as a project leader.

The training using the process was conducted at the university. In the training, the main concern was to explain carefully the RiPLE-SC process since all subjects had previous SPL knowledge.

6.4.3. Data Collection Procedures

According to Lethbridge et al. (2005), data collection techniques can be divided into three levels:

- **First degree**: Direct methods mean that the researcher is in direct contact with the subjects and collect data in real time. This is the case for interviews, focus groups, and observations.

- **Second degree**: Indirect methods where the researcher directly collects raw data without actually interacting with the subjects during the data collection. This approach is indicated, for example, in scenarios where the use of software engineering tools is automatically monitored, and observed through video recording.

- **Third degree**: Independent analysis of work artifacts where already available and sometimes compiled data is used. This is the case when documents such as requirements specifications and failure reports from an organization are analyzed or when data from organizational databases such as time accounting is analyzed.

Applying the techniques, the data collect should be based on the point of view point of different sources. It is important to use several data sources in a case study in order to limit the effects of interpretation from one single data source (Runeson and Host, 2009). Thus, this case study aimed to evaluate the RiPLE-SC from the perspective of different roles into process. In this sense, measurement instruments were used for data collection, such as questionnaires, that evaluate the background of the subjects on SPL (especially on SPL scoping) and the subjects’ satisfaction, guidance and difficulties using the proposed process, as well as the effectiveness of the process. All questionnaires used in this case study can be seen in Appendix D.
In order to have more confidence for the data collected, especially, data related to effort, the subjects will be oriented to use a timesheet to register the time elapsed while performing the steps of the process.

6.4.4. Measures
The data collected in an empirical study may be quantitative or qualitative. Quantitative data involves numbers and classes, while qualitative data involves words, descriptions, pictures, diagrams, and so on (Runeson and Host, 2009). In this sense, in this case study we used a combination of qualitative and quantitative data to provide a better understanding of the studied phenomenon (Seaman, 1999).

In this context, criteria were defined in order to evaluate the benefits using of the RiPLE-SC. The benefits were evaluated using as base, metrics related to understandability, effort, completeness and effectiveness.

6.5. Analysis and Results Reported
After defined the right context for the RiPLE-SC process use, a preliminary schedule for the project was negotiated between the head of RiSE Labs and the director of the MedicWare in a meeting in December 20, 2009 (with duration of thirty minutes). In this meeting, it was defined that the scope would be defined based on the four main products developed in the company (See the products description in Section 4.1).

6.5.1. Execution of the RiPLE-SC
In the case study the four phases of the RiPLE-SC were executed. In the next sections the execution of the case study is presented in details.

6.5.1.1. Pre-Scoping
With the negotiation of the context and schedule, the application of the RiPLE-SC started in January 04, 2010 with the phase pre-scoping. This phase was initiated with the task pre-scoping meeting. The meeting was performed at the MedicWare (with duration of one hour and nineteen minutes) and conducted by the scoping expert, author of this dissertation. Appendix E presents all the information related to meetings. It is important highlight that the audio of all the meetings were recorded.
In this meeting, a general view of SPL was presented, as well as, a quick training in SPL scoping was performed with the exhibition of the RiPLE-SC process. After the introduction on SPL and the RiPLE-SC, a questionnaire (See Appendix D) was applied to the members of the company which participated in the meeting: the manager (who was a software development leader in the company for the desktop products) and the business analyst. It is important to highlight that this questionnaire was also answered by the main architect/developer (technical leader) and by the market expert of the company. The questionnaire aimed to gather information about the operational and organizational context of the company, the profile of their stakeholders and the expectations with the project. Thus, it was possible to identify the following aspects: the stakeholders do not have knowledge of SPL and perform several roles in the company, the main expectations with the project are to increase the productivity and quality, and decrease effort and costs, the company does not have a formal development process, and the products documentation is available only in a wiki system, where is presented a training about them, i.e., there are not requirements and tests documentation. Moreover, the products are evolutionary, i.e., they are constantly improved with new features requested by customers and their sales are performed by a set of modules available for selling not by features selection.

In this context, with the pre-scoping meeting, we defined the stakeholders that would participate of the project and their roles according to the ones defined in the RiPLE-SC process (See Section 5.3.3) and concluded that the scope definition could aid in the internal organization of the company, as well as, improving the documentation process, besides, making possible a better Return on Investment (ROI) with the products redefinition based on a set of features. Table 6.1 shows the roles presented in the RiPLE-SC and the association with the stakeholders of the MedicWare and RiSE in the case study.

It is possible to see in Table 6.1 that the manager of the MedicWare is associated at the roles of customer and domain expert, moreover, the company developers that also perform the role of architect are associated to the roles of developer, architect and domain expert in the case study. It is also possible to see that the role domain expert is performed by the business analyst of the MedicWare and by the project team from RiSE. The RiSE project team is also
associated to the roles of market expert, developers and architect in the case study. Besides, the role scoping expert is associated to the author and the co-advisor of this dissertation. This occurs because the case study is composed of two distinct teams working together: MedicWare and RiSE. Other factor is related to the rotative aspect of the team since some members worked virtually, from another city in some periods in the project.

Table 6.1 Stakeholders and roles of the case study

<table>
<thead>
<tr>
<th>Role: RiPLE-SC</th>
<th>Role: MedicWare</th>
<th>Role: RiSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Line Manager</td>
<td></td>
<td>Project Manager (co-advisor of this dissertation) and the Project Leader</td>
</tr>
<tr>
<td>Scoping Expert</td>
<td></td>
<td>Scoping Expert (Author of this dissertation) and the RiSE Project Manager (co-advisor of this dissertation)</td>
</tr>
<tr>
<td>Customer</td>
<td>Project Manager</td>
<td></td>
</tr>
<tr>
<td>Market Expert</td>
<td>Market Expert</td>
<td></td>
</tr>
<tr>
<td>Developer</td>
<td>Developer/Architect (three rotative members)</td>
<td>Project team (seven rotative members)</td>
</tr>
<tr>
<td>Architect</td>
<td>Developer/Architect (three rotative members)</td>
<td></td>
</tr>
<tr>
<td>Domain Expert</td>
<td>Project Manager, the Business Analyst and the developers</td>
<td></td>
</tr>
</tbody>
</table>

The pre-scoping phase is also composed of the task analyze market. In the case study, the information associated to the market was obtained across the analysis of the documentation. The contact with the market expert was very important since he had a lot of material about it.

In the meeting with the market expert were discussed the following issues: the marketing investments made by the company, the product market leader and possible reasons for it, the main competitors and innovation aspects. Thus, we identified that the MedicWare invests few in marketing programs, factor that hinders the dissemination of their products in the market. According to the market expert, even they do not have a product market leader, their products have high quality and are able to compete with other ones. We also observed that the products were developed and evolved from customers’
requests and not by the company innovation. With the identification of these aspects, a business plan was created to document the information gathered.

6.5.1.2. Domain Scoping

After the identification of the market aspects related to MedicWare and their documentation, it was initiated the domain scoping phase. The domain scoping had as objective to identify the domains and sub-domain with more potential to compose the products of the product line. This phase initiated with the evaluation of the products documentation. This evaluation was performed by the RiSE members at the Federal University of the Bahia.

However, the results obtained were not satisfactory, because the documentation offered by the company was inconsistent and incomplete without any requirements specification. Nevertheless, it was possible to define the domains initial list which would be analyzed in the task analyze domains that compose the phase domain scoping. Thus, the product line manager contacted the MedicWare manager for that a meeting was scheduled, whose focus would be discuss the domains identified. This meeting marked the beginning of the task analyze domains, that is conducted through a workshop to enable the knowledge dissemination among the stakeholders and make possible the collaborative work. Three workshops, at the company were performed in the task analyze domains.

This task is composed of four steps: review domains, identify sub-domains, analyze sub-domains and prioritize sub-domains. In this sense, in the first workshop were executed the two first steps of this task.

In the domains review, the manager and the business analyst of the MedicWare explained the four domains initially identified, showing their main functionalities, making possible the stakeholders understanding on them. After this understanding, it was defined that those would be the domains that would compose the product line, i.e., the domains initial list was maintained. The explanation of the MedicWare’s members about the domains favored the posterior identification of the sub-domains. In this context, the choice of the sub-domains was based on the modules which integrate the MedicWare’s products. Thus, for each module existent was associated one sub-domain. At the final of this step, forty-one sub-domains were discussed and identified. Still in
the first workshop, the manager of the MedicWare if compromised in supplying the updated version of the existent documentation.

In the second workshop of the task analyze domain, the step analyze sub-domains was performed starting with a brief explanation of the scoping expert on how the workshop would be conducted. The scoping expert explained that the sub-domains would be analyzed according to the following aspects: experience, risk, volatility, maturity, potential from the available code, reuse potential, market potential and coupling. Besides, the stakeholders were oriented for keeping the focus on the analysis. The definition for the used aspect (experience, maturity, and so on) can be seen in Chapter 5, section 5.4.2.1.

In this workshop, nineteen sub-domains were analyzed. With the use of the workshop, we could see the effective collaboration among the stakeholders. Moreover, the workshop was important to the RiSE’s members to gather detailed information about the sub-domains, facilitating the execution of the others phases of the RiPLE-SC process.

In the last workshop, of the task analyze domains, twenty-two sub-domains were discussed, finalizing this step.

As the availability of the MedicWare’s stakeholders was limited and the information volume obtained in the workshops was high, the last step of the task analyze domains, i.e., the step prioritize sub-domains was not performed in the workshop format in this case study. Moreover, as the MedicWare had few documentation (a wiki system), we decided to document the information identified in the sub-domains analysis. In order to perform it, the scoping expert listened the audio from the workshops and extracted the information. Then, all the information was documented according to the template that can be seen in Appendix C.

After the documentation, the sub-domains were analyzed and prioritized by the scoping expert. The goal was to identify the sub-domains more able to increase the company market potential. Based on this analysis from forty-one initials sub-domains, one was discarded, three were characterized as features and six were defined as reusable, the others ones were also chosen to compose the product line.
6.5.1.3. Product Scoping

According to the RiPLE-SC process, after the phase domain scoping, is initiated the product scoping. This phase is composed of the tasks: construct user stories, identify features, features review meeting, identify products, construct product map and validate product map. However, in the case study, the task construct user stories was not necessary, because the project was characterized as a reengineering one, thus, the analysis of the documentation available at the company was the most indicated way to be used as basis for the features identification.

In this context, the task identify features started with the documentation analysis. This step was performed across a collaborative work among all members from RiSE. In it, each member became responsible for the analysis of a sub-domain group. However, this analysis was very hard to perform, in consequence of the incomplete documentation offered by the company. This step was finished with the identification of 3644 features.

After the analysis, we identified the need of a domain glossary, mainly, in this context where the project stakeholders (RiSE) do not have domain knowledge. This artifact is defined in the RiPLE-RE, the Requirements Engineering process of the RiPLE. However, we identified that, when necessary, it should be initially specified during the RIPLE-SC. It is important to highlight that it should be maintained during all project.

One problem found during the features identification was the granularity level that would be used. However, this problem was mitigated with workshops among the RiSE members to validate the features initially identified.

In these workshops, features which cause agreement difficulties during the documentation analysis were exhaustively discussed from different perspectives, since the stakeholders were stimulated to express their perspectives about them. At the end of the workshops, two thousands nine hundred and forty features were discarded (2940).

As the analysis for identification of the features was performed based on an incomplete documentation and since no one member from RiSE was a domain expert, the RiSE product line manager requested some training sessions from MedicWare, in order to improve the sub-domain understanding and feature identification.
The trainings were performed in three meetings conducted by one of the MedicWare expert users in the products. Three RiSE Members also participated and it was considered extremely useful since in the end of the trainings, more than one hundred new features (110) were identified.

In parallel with the task identify features, was performed the task features review meeting. The main goal of this task was to get feedback from customers about the features identified and make a common sense on it. During fifty days (January 20, 2010 – February 03, 2010) one member from MedicWare and a validation committee from RiSE worked together on it.

The customer validation, internal workshops and the training sessions were important filters to develop the final feature list. At the end of the tasks, identify features and features review meeting, a list composed of eight hundred and forty features (840) was consolidated.

After the feature list consolidation, its description was started. As it is a laborious task and based on the high number of features, we prioritized the features more complex to understand in the domain.

According to the RiPLE-SC process, after the features definition, the products definition should start. However, as the scope identified in this case study is related to available products, the choice of the products for composing the product line was based on the MedicWare’s four products.

Based on the list of features and products, the product map was developed. It is used for relating products and their features, making possible to obtain a general vision of the product line and facilitate its understanding. After the development of the product map, the task validate product map was performed. In this task, participated three RiSE members and the manager and the business analyst from MedicWare. The main goal of this task was: 1. identify if any features was forgotten or allocated improperly in some product; and 2. verifying the scope defined for the features, i.e., if they would be, in fact, mandatory, optional or out of scoping. In this validation, the final product map was not modified.

With the execution of the phase product scoping, it was possible to identify that some aspects are highly important for the good course of the tasks of a scoping process. The lack of a domain expert from the RiSE, with effective knowledge in the medical domain, the lack of a complete and consistent
documentation about the domains and sub-domains identified and the time unavailability from the MedicWare members caused rework in several steps. Moreover, the time unavailability was also responsible for difficulties to schedule meetings at the company. All these aspects influenced negatively in the execution agility of the RiPLE-SC. However, even with these problems found, the artifacts developed were satisfactory, corresponding the real needs of the customer.

6.5.1.3. Assets Scoping

The last phase of the RiPLE-SC process is the assets scoping. It is composed of three tasks: create metrics, apply metrics and prioritize product map. The definition and application of metrics is performed for enabling the choice of the final assets set that should compose the product line and in the end of this phase the product map is prioritized based on the metrics results and customers' needs.

The task create metrics is based on the business goal defined by the project stakeholders during the task pre-scoping meeting, according to the GQM model (See Section 5.4.4). In this context, this task started with research performed by the scoping expert to identify consolidates metrics associated to the business goal defined by the stakeholders. However, in consequence of the difficulty for finding these metrics and time constraints of the project, we decided to prioritize the product map considering only the goal decrease costs and metrics related to it. Thus, the following metrics were defined: cost, complexity and aggregate value. The Tables 6.2, 6.3 and 6.4 show the metrics defined in this case study according to the GQM model.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Enable the costs decrease for the product from the point of view of the manager.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questions</td>
<td>1. What is the development cost for the feature?</td>
</tr>
<tr>
<td>Characterization Metrics</td>
<td>- Cost of $f$ in the product (Levels: high, medium and low)</td>
</tr>
<tr>
<td></td>
<td>cost($f$)</td>
</tr>
<tr>
<td>Benefit Metrics</td>
<td>What is the cost to define the feature $f$ for the product?</td>
</tr>
<tr>
<td></td>
<td>$C(f) = cost(f)$</td>
</tr>
</tbody>
</table>

Table 6.2 Cost metric
Table 6.3 Complexity metric

<table>
<thead>
<tr>
<th>Goal</th>
<th>Decrease the product development complexity from point of view of the developer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questions</td>
<td>1. How complex is the feature implementation?</td>
</tr>
<tr>
<td>Characterization</td>
<td>- Complexity to implement $f$ (Levels: high, medium and low)</td>
</tr>
<tr>
<td>Metrics</td>
<td>complexity($f$)</td>
</tr>
<tr>
<td>Benefit Metrics</td>
<td>What is the complexity associated with $f$ implementation in the product?</td>
</tr>
<tr>
<td></td>
<td>Comp($f$) = complexity($f$)</td>
</tr>
</tbody>
</table>

Table 6.4 Aggregate value metric

<table>
<thead>
<tr>
<th>Goal</th>
<th>Enable the economical potential increase from the point of view of the market expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questions</td>
<td>1. What is the aggregate value for the feature in the product?</td>
</tr>
<tr>
<td>Characterization</td>
<td>- aggregate value for the feature $f$ (Levels: high, medium and low)</td>
</tr>
<tr>
<td>Metrics</td>
<td>aggregatevalue($f$)</td>
</tr>
<tr>
<td>Benefit Metrics</td>
<td>What is the aggregate value with the implementation of the feature $f$ in the product?</td>
</tr>
<tr>
<td></td>
<td>AV ($f$) = aggregatevalue($f$)</td>
</tr>
</tbody>
</table>

With the metrics defined and the product map constructed, the task apply metrics was performed. The metrics were applied by the scoping expert based on estimative obtained from the MedicWare manager, developer and market expert. After this task, the product map was prioritized and one hundred and forty-seven (147) features were modified in relation to its scope (mandatory, optional or out of scope). It involved the MedicWare manager, the RiSE manager and others three RiSE members.

Finished the phase assets scoping, one additional step was defined with the goal of obtaining the feedback from RiSE and MedicWare members about the project.
Initially, we performed one postmortem meeting with the RiSE members following the guidelines defined in Collier et al. (1996). In this meeting the participants discussed during two hours, aspects such as: faults, difficulties, as well as forms of mitigate them. These aspects are presented in the topic lessons learned (See Section 5.3). Moreover, in order to evaluate the process, one questionnaire (See Appendix D) was applied for four RiSE members that participated full time in the project.

The feedback from MedicWare was obtained after the meeting for the project final presentation. It had the participation from RiSE members and the MedicWare manager. In this meeting, the following points were discussed:

- **Satisfaction of the company with the project.** The enterprise believe in the success of the project and that the scope identified will decrease costs and increase the productivity in the enterprise;

- **Communication forms.** According to the MedicWare, the communication forms used during the RiPLE-SC were satisfactory in all the steps;

- **Importance and quality of the artifacts generated.** The company did not have how to measure the quality level of the artifacts. However, in consequence of the feedbacks and the effective communication, it believes that the artifacts were defined with quality and will be fundamentals for the success of the SPL.

This meeting was the final landmark for the case study. Figure 6.3 shows the timeline for the project.

![Figure 6.3 Timeline of the project phases](image)

### 6.5.2. Data Validation

In the case study, the RiPLE-SC was instantiated by seventeen members, including six ones from MedicWare and RiSE. However, the data were collected
only from four RiSE members (See Table 6.5), because the others did not participate full time in the case study.

Analyzing the members’ profile, it is possible to conclude that they have participated in some industrial and academic projects, they had low experience in SPL activities, especially, in industrial projects. Regarding to scoping, the majority has the academic knowledge, but low industrial practice. Table 6.5 shows a summary of members’ profile.

**Table 6.5 Members’ profile**

<table>
<thead>
<tr>
<th>ID</th>
<th>Degree</th>
<th>Participation in Industrial Projects</th>
<th>Participation in Academic Projects</th>
<th>Experience in SPL</th>
<th>Experience in Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Master Candidate</td>
<td>1, low complexity</td>
<td>1, low complexity</td>
<td>Academic-medium</td>
<td>Academic-medium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1, medium complexity (in SPL)</td>
<td>2, medium complexity (1 in SPL)</td>
<td>Industrial-medium</td>
<td>Industrial-medium</td>
</tr>
<tr>
<td>2</td>
<td>P.H.D. Candidate</td>
<td>2, low complexity</td>
<td>______</td>
<td>Academic-medium</td>
<td>Academic-low</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2, medium complexity</td>
<td></td>
<td>Industrial-low</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Master Candidate</td>
<td>______</td>
<td>1, low complexity (in SPL)</td>
<td>Academic-medium</td>
<td>Academic-low</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1, medium complexity</td>
<td>Industrial-low</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Master Candidate</td>
<td>3, high complexity</td>
<td>1, low complexity (in SPL)</td>
<td>Academic-medium</td>
<td>Academic-medium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1, medium complexity (in SPL)</td>
<td>Industrial-low</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1, high complexity (in SPL)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**6.5.3. Data Interpretation**

After collecting data from case study, it is important to draw conclusions based on it as follows.

**6.5.3.1. Process Understanding and Application Difficulties**

Analyzing subjects’ answers for the understanding and difficulties in the use of the RiPLE-SC process, it was identified that only one subject had some problem in the tasks. This subject (ID 2) had difficulties related to features identification, especially, the right granularity level that a specific feature should have. These difficulties can have occurred in function of the experience lack that the subject
presents in SPL projects since does not participated before of projects in this area, as can be seen in Table 6.5.

As only one subject reported to have some difficulty in the RiPLE-SC, the null hypothesis ($H_01$: $\text{PUAD} \geq 40\%$) was rejected.

### 6.5.3.2. Satisfaction

Regarding subjects satisfaction, 50% (ID 1,3) of them were *very satisfied* and 50% (ID 2,4) were *satisfied* using the process.

The subjects 1 and 3 affirmed that the process guidelines make possible a complete analysis of the domains enabling to identify the most suitable ones. Moreover, they believe that the agile practices used in the process aid in the execution agility of the tasks.

The subject 2 presented some difficulties during the project, mainly, in the features identification, being this, the main factor for it to indicate its satisfaction level as satisfied. However, this subject did not have SPL experience and their difficulties can have been occasioned by maturity lack in SPL.

The subject 4 indicated its satisfaction level as satisfied, because it believes that the process should present a solution more consolidated for the metrics used in assets scoping.

Figure 6.4 presents the frequency of the subjects satisfaction according to the defined scale.

![Figure 6.4 Satisfaction level](image)

### 6.5.3.3. Guidance

Analyzing the answers related to the guidance for application of the RiPLE-SC, we could identify that one subject (ID 2) reported that in the most of the time, there was the guidance. On the other hand three of them (ID 1, 3, 4) affirmed that sometimes this happened. However, in spite of present distinct answers, all
the subjects justified that the main factor related to the lack of guidance was that during the execution of the tasks the process expert more appropriated was not available, because during the project the process expert had to work remotely from another city.

### 6.5.3.4. Effort to Apply each Phase of the Process (EAPP)

In this case study, 740 hours and 58 minutes were spent including all effort associated to the four phases of the RiPLE-SC, as can be seen in Figure 6.5. Appendix E presents the whole timesheet for the project.

#### Figure 6.5 Effort distribution during the project

Analyzing the data set, it is possible see that product scoping was the most extensive phase. We believe that it happened because the high number of features identified. Moreover, the problems faced with the company documentation (inconsistency, incomplete) contributed for this scenario. Figure 6.6 shows the effort percentage for each phase of the RiPLE-SC.

#### Figure 6.6 Scoping percentage for each phase of the RiPLE-SC in the case study
As we did not have any baseline for comparison and our intention was define and characterize it for future projects, we did not define any hypothesis for this metric.

6.5.3.5. Tasks, Roles and Artifacts Missing

The idea behind this question was to collect more information about the RiPLEySC missing tasks. In this sense, the subjects were asked if there was any missing task.

Analyzing the answer from the subjects, no one identified any missing task, rejecting the null hypothesis ($H_{02}$: TRAM $\geq 2$). Nevertheless, all the subjects had previous knowledge from SPL and some of them had participated in industrial (25%) or academic (75%) SPL project.

6.5.3.6. Effectiveness

Analyzing the effectiveness of the agile practices adopted by the RiPLE-SC, it was possible to identify that the process was effective in its purpose. All the subjects defined as good, the practices adopted, using a scale ranging from excellent to terrible. According to the analysis, the factor that presented problem in the agility of the RiPLE-SC execution was the step feature description. However, this step is optional and it is indicated in domains where the team does not have maturity in them or in situations where the features name can bring problems related to ambiguity. One important point to analyze is the subject background on agile practices. In our questionnaire, we did not collect this information and we have to update it for future projects, since it could influence the analysis.

6.5.4. Lessons Learned

After concluding the case study, some aspects should be considered in order to replicate it in others contexts.

**Initial Planning.** During the case study some problems occurred in consequence of the lack of initial planning. This had negative impact, for example, in the identification of the domains and features, because the documentation related to company products was availability in parts and with
inconsistence. With a initial planning, where the documentation solicitation was performed before of starting the project;

**Documenting Experiences.** In the beginning of the project, discussions about management tools such as, Project, XPlanner, as well as others ones were performed generating unnecessary effort. Considering that these discussions were also performed in others projects, they could be reused.

**Experts Availability.** It is very important to have the scoping and domain expert actively participating in the project. In the project, the lack of a domain expert in the RiSE team and the unavailability of the MedicWare domain experts contributed with problems in the features identification resulting in a lot of rework. Moreover, as the scoping expert (author of this dissertation) participated remotely in some tasks, the execution and management was pledged, in some situations.

**Validation.** In general, the experts availability is limited, thus, scheduling meetings is more difficult. In this context, a solution found during the project was to send some artifacts by email for experts’ evaluation and subsequently to schedule one meeting only for explanation.

**The Role of Domain Expert.** During the project the team perceived the need of a domain expert with technical knowledge on the domains, i.e. with experience in the development of features related to the domains. It could facilitate and make more productive the feature identification task.

**Features Documentation.** The features documentation is an arduous task. Therefore, they should be described in an incremental way during the lifecycle of the SPL.

**Granularity of the Features.** It is very important that the project team knows the limit between features and requirements. The features should be defined in high level and the understanding difficulty in this concept can bring rework for the project.

**Identify Metrics of the Customers.** An important aspect that should be collected is the effort of the customer during the project. The main goal behind this aspect is to identify not only the effort of them in the workshops and meetings.

**Glossary.** The need of a domain glossary is very important, mainly, in contexts where the project stakeholders do not have domain knowledge. This artifact is
defined in the RiPLE-RE, the Requirements Engineering process of the RiPLE. However, we identified that, when necessary, it should be initially specified during the RiPLE-SC.

**Support Tool.** The use of a specific support tool is essential for subjects’ productivity, reducing rework in the scoping tasks, mainly, in tasks related to features identification and development and management of the product map.

### 6.6. Chapter Summary

This chapter presented the definition, planning, analysis and report of the results of an industrial case study to evaluate the RiPLE-SC process. The study analyzed it from the perspective of researchers on the process understandability, satisfaction, guidance, effort and effectiveness for the context of a software product line project in a small company.

The qualitative analysis indicated a good potential according to subjects satisfaction and that the process is effective in its agile purpose. Other aspect that presented good results was the guidance. Its analysis indicated that the documentation associated to the process presents tasks and guidelines well-defined. Regarding to the quantitative analysis, the case study indicated that, 75% of the subjects did not have difficulties in applying the process. Moreover, it showed that the process presents the essentials tasks for a scope definition with quality. However, analyzing the benefits and trade-offs of the proposed process, more case studies should be conducted in the future with different companies and environments.

Next chapter presents the conclusions of this work, its main contributions and directions for future work.
The frequent changes in the software industry and the search to decrease effort, costs and time-to-market and increase the productivity and quality make the agile software product lines an attractive approach.

In this context, agile scoping can be a good way for decreasing the planning costs up-front and, consequently, make possible to decrease the time-to-market.

Based on this scenario, we proposed a new process which is defined in a systematic and agile way. The proposed process was evaluated in a case study, which presented the benefits for using it and directions for improvements.

In this chapter, we present research contributions, related work, future work, academic contributions and the concluding remarks of this dissertation.

7.1. Research Contributions
The main contributions of this work can be split into the following aspects: 1. systematic review on scoping for software product lines; 2. the definition and systematization of an agile scoping process for software product lines; 3. the execution of an industrial case study which evaluated the proposed process. These contributions are further described next.

- **Systematic review on scoping approaches for Software Product Lines.** Through this review, eleven approaches were identified and analyzed according to the aspects related to the review research questions. The analysis results were used to guide the definition for the scoping process for SPL. 
• **RiPLE-SC.** After the systematic review and the study on agile methods, their results were used as inputs for the definition of RiPLE Scoping process (RiPLE-SC). It is defined in a systematic manner, with tasks, inputs, outputs, roles and guidelines addressed to the context of software product lines. Besides, it inserts agile practices and techniques in the execution of their tasks, making possible to decrease up-front planning time and, consequently, project costs.

• **Case Study.** A case study was applied in an industrial environment in order to evaluate the proposed process. This initial validation of RiPLE-SC aided to improve the process, since the findings suggested some modifications in the tasks description to clarify the tasks understanding and decrease the effort during the execution of the process.

### 7.2. Related Work

Ten scoping approaches for SPL and one approach of agile scoping for SPL were analyzed through the systematic review, as described in Chapter 3. However, among the approaches included in this review, there are many gaps in the scoping activities. In general, there is not a complete scoping process for SPL and the agile research in this area is beginning, as described in Chapter 4. Thus, the output of the systematic review and the overview about agile SPL served as a foundation for the RiPLE-SC definition.

### 7.3. Future Work

Due to the time constraints imposed on a M.Sc. degree, this work can be seen as an initial step towards the efficient, usable and effective agile scoping process in the context of software product lines. Thus, there are interesting topics to improve what has been started and new paths to explore. Thus, the following issues should be investigated as future work:

• **Application of RiPLE-SC in others industrial projects.** This dissertation presented the definition, planning, execution and results of a case study. However, new case studies are necessary in conjunction with the industry, in different contexts, in order to refine the process.

• **Support tool.** The use of an appropriate support tool is essential to efficiency of a scoping process. Thus, it shows the need of a tool to
support tasks as document features and define the product map, making possible the integration of these tasks.

- **Metrics.** The use of metrics to aid as filter has relevant potential in the scope definition. However, lack metrics consolidated, thus, this shows the need of studies more detailed in this field.

- **Cost Models.** The cost models definition is important to indicate the economical potential of product lines. These models can be used to define several costs types such as core assets costs, product development costs. In this sense, this shows the need of an elaborated cost model to aid in the definition of the most suitable scope for SPL.

### 7.4. Academic Contributions

As an intermediate result of the work presented in this dissertation, the following contributions can be enumerated:

- (Moraes et al., 2009) A Systematic Review on Software Product Lines Scoping;

### 7.5. Concluding Remarks

This work presented an agile scoping process for software product lines, which can be seen as a systematic and agile way to define scoping. This process was based on an extensive systematic review on scoping, whose results can be used for further research on scoping as well as in agile product lines.

Additionally, a case study was conducted to evaluate the effort, effectiveness, completeness, understandability, satisfaction and guidance of the RiPLE-SC process. The qualitative analysis indicated a good potential according to subjects satisfaction and that the process is effective in its agile purpose. Other aspect that showed good results was the guidance. Regarding to quantitative analysis, the case study indicated that 75% of the subjects did not have difficulties in applying the process. Besides, it showed that the process presents the essentials tasks for a scope definition with quality.

Finally, we believe that the combination of SPL and AM is a topic that can bring a significant impact and benefits to the software industry and the RiPLE-SC can be viewed as a starting point to achieve these goals.


Product Line Engineering, collocated with the International Software Product Line Conference, Baltimore, MD, USA.


Discovered Knowledge Techniques, Brazilian Symposium on Software Components, Architectures, and Reuse, Porto Alegre, Brazil.


Appendix A. Systematic Review Quality Criteria

This appendix presents the quality criteria used to select the studies in the systematic review procedure detailed in Chapter 3. The evaluation results are also presented in this Appendix.

Table A.1 presents the quality criteria for the scoping approaches of SPL.

<table>
<thead>
<tr>
<th>Quality Criteria</th>
<th>Options</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are the metrics defined?</td>
<td>Yes. They are explicitly defined in the study.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Partly. They are implicit.</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>No. They are not defined.</td>
<td>0</td>
</tr>
<tr>
<td>Are the guidelines described?</td>
<td>Yes. They are explicitly described for all activities found in the study.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Partly. They are described for some activities found in the study.</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>No. They are not described for all activities found in the study.</td>
<td>0</td>
</tr>
<tr>
<td>Are the roles defined?</td>
<td>Yes. They are defined.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>No. They are not defined.</td>
<td>0</td>
</tr>
<tr>
<td>Are the approaches experimented?</td>
<td>Yes. They are experimented in the industry.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Yes. They are experimented in the academic.</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>No. They are not experimented.</td>
<td>0</td>
</tr>
<tr>
<td>Are the stakeholders defined?</td>
<td>Yes. The stakeholders are defined.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>No. The stakeholders are defined.</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>No. The scope is not evaluated.</td>
<td>0</td>
</tr>
<tr>
<td>Is the scope optimized?</td>
<td>Yes. The scope is optimized.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>No. The scope is not optimized.</td>
<td>0</td>
</tr>
<tr>
<td>How well is the study detailed?</td>
<td>High. Study is well detailed and it is easily understood.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Medium. Study is little detailed, but it can be understood.</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Low. Study presents an insufficient detail level to its understanding.</td>
<td>0</td>
</tr>
</tbody>
</table>
The results of the application of these quality criteria are presented in Table A.2. This result shows an overview about the quality score of the studies. However, it is difficulty establishing a reliable relationship between final quality score and the real quality of each study.

<table>
<thead>
<tr>
<th>Approaches Author</th>
<th>Metrics</th>
<th>Guidelines</th>
<th>Roles</th>
<th>Experiment</th>
<th>Stakeholders</th>
<th>Optimization</th>
<th>Detail Level</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Riebisch [2001]</td>
<td>NO</td>
<td>PARTLY</td>
<td>NO</td>
<td>NOT INFORMED</td>
<td>NO</td>
<td>YES</td>
<td>MEDIUM</td>
<td>2.0</td>
</tr>
<tr>
<td>Kish [2002]</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>IN INDUSTRY</td>
<td>NO</td>
<td>NO</td>
<td>HIGH</td>
<td>3.0</td>
</tr>
<tr>
<td>Schmid [2002]</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>IN INDUSTRY</td>
<td>NO</td>
<td>YES</td>
<td>HIGH</td>
<td>5.0</td>
</tr>
<tr>
<td>Rommes [2003]</td>
<td>NO</td>
<td>PARTLY</td>
<td>NO</td>
<td>NOT INFORMED</td>
<td>YES</td>
<td>NO</td>
<td>MEDIUM</td>
<td>2.0</td>
</tr>
<tr>
<td>Lee [2004]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helferich [2005]</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>IN ACADEMY</td>
<td>YES</td>
<td>YES</td>
<td>MEDIUM</td>
<td>4.0</td>
</tr>
<tr>
<td>Park [2005]</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>IN ACADEMY</td>
<td>NO</td>
<td>NO</td>
<td>HIGH</td>
<td>3.5</td>
</tr>
<tr>
<td>Clements [2005]</td>
<td>NO</td>
<td>PARTLY</td>
<td>NO</td>
<td>IN INDUSTRY</td>
<td>YES</td>
<td>NO</td>
<td>LOW</td>
<td>2.5</td>
</tr>
<tr>
<td>John [2006]</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>IN INDUSTRY</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>6.0</td>
</tr>
<tr>
<td>Inoki [2007]</td>
<td>YES</td>
<td>YES</td>
<td>PARTLY</td>
<td>IN ACADEMY</td>
<td>NO</td>
<td>NO</td>
<td>HIGH</td>
<td>4.0</td>
</tr>
<tr>
<td>Noor [2007]</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>IN INDUSTRY</td>
<td>YES</td>
<td>YES</td>
<td>HIGH</td>
<td>5.0</td>
</tr>
</tbody>
</table>
Appendix B. Data Collection Template for the Systematic Review

DATA COLLECTION FORM
A Systematic Review on Scope Definition Processes for Software Product Lines

APPRAISER: | APPRAISAL DATE:
---|---

PUB. TITLE:

REVIEWER: | COMPLIANCE: [] OK [] NOT OK
---|---

[] INCLUDED | [] NOT INCLUDED

CRITERIA: [Fill in this blank with explanation about the criteria this paper fits in – EXCLUSION OR INCLUSION]

RESEARCH QUESTIONS
[Which and How this paper answer the following?]

Q1. What activities are addressed for scope definition in the software product lines approaches and processes?

Q2. Do the SPL approaches optimize scope?

Q3. What are the types of scope used by the approaches?
Q4. Which are the stakeholders involved in the scope definition process?

Q5. Do the approaches use specific metrics or cost models for scope definition?

Q6. Are the approaches customizable?

Q7. Do the approaches treat the new perspective of agile SPL planning?

Q8. How are the approaches related to SPL development?

<table>
<thead>
<tr>
<th>QUALITY CRITERIA ASSESSMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>QC1</td>
</tr>
<tr>
<td>QC2</td>
</tr>
<tr>
<td>QC3</td>
</tr>
<tr>
<td>QC4</td>
</tr>
<tr>
<td>QC5</td>
</tr>
<tr>
<td>QC6</td>
</tr>
<tr>
<td>QC7</td>
</tr>
</tbody>
</table>

TOTAL
Appendix C. RiPLE-SC Templates

In this appendix, the templates of RiPLE-SC will be described and detailed.

C.1 Template for Sub-Domain Analysis

| Domain Name: |
| Sub-Domain Name: | Description: |
| Experience: | Comments: |
| Risks: | Comments: |
| Volatility: | Comments: |
| Maturity: | Comments: |
| Existent Code Potential: | Comments: |
| Market Potential: | Comments: |
| Reuse Potential: | Comments: |
| Coupling | Comments: |

C.2 Template for Listing Features

| Feature Name | Description | Priority |

C.3 Template for Mapping Features and Products

<table>
<thead>
<tr>
<th>Features</th>
<th>Sub-Features</th>
<th>Sub-Sub-Features</th>
<th>Product</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>P1</td>
<td>P2</td>
</tr>
</tbody>
</table>
## C.4 Template for Mapping Features and Products

<table>
<thead>
<tr>
<th>Objective</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Questions</td>
<td></td>
</tr>
<tr>
<td>Characterization</td>
<td></td>
</tr>
<tr>
<td>Benefit</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D. Case Study Questionnaire

As part of the case study, detailed in Chapter 6, two questionnaires were defined, and applied.
The first one (detailed in Section D.1) was intended to collect data about the stakeholders as well as information such as organizational structure and culture, aspects identified in the pre-scoping meeting. The second one (detailed in Section D.2) was intended to collect information about the RiPLE-SC uses and the background of the subjects that answered the questionnaire.

D.1 QT1 – Pre-Scoping Questionnaire

Date:
Name:

**Degree:** ( ) Graduate ( ) M.Sc. ( ) Ph.D ( ) Specialization

**Which role best describes your current position in your company?**
- [ ] President
- [ ] Development Manager
- [ ] Product Manager
- [ ] Requirements Analyst
- [ ] Team Lead
- [ ] Tester
- [ ] Domain Analyst
- [ ] Domain Expert
How many years of experience do you have in the function which performs?

___________________________________________________
___________________________________________________
___________________________________________________

How could you classify your experience in this domain?
☑ Excellent
☑ Good
☑ Medium
☑ Low

Do you know any reuse techniques? Which one(s)?
Please cite the techniques and define your knowledge level according to the following scale: excellent, good, medium, low.

___________________________________________________
___________________________________________________
___________________________________________________

How many projects using some reuse technique have you participated?

Large complexity:
☑ None ☐ 1 - 2 ☐ 3 - 7 ☐ More than 7

Medium complexity:
☑ None ☐ 1 - 2 ☐ 3 - 7 ☐ More than 7

Small complexity:
☑ None ☐ 1 - 2 ☐ 3 - 7 ☐ More than 7

Do you know any software product lines techniques? Which one(s)?
Please cite the techniques and define your knowledge level according to the following scale: excellent, good, medium, low.

___________________________________________________
___________________________________________________
___________________________________________________
How many projects using some product lines technique have you participated?

Large complexity:
- None
- 1 - 2
- 3 - 7
- More than 7

Medium complexity:
- None
- 1 - 2
- 3 - 7
- More than 7

Small complexity:
- None
- 1 - 2
- 3 - 7
- More than 7

Do you know any product lines scoping practices? Which one(s)?
Please cite the practices and define your knowledge level according to the following scale: excellent, good, medium, low.

What are your expectations with this project?
- Reduce Time-to-Market
- Reduce development effort
- Improve Productivity
- Improve Quality
- Increase the Company Portfolio
- Reduce Costs
- Gain New Markets
- Other [    ]

How is the current organizational structure of the company?

What is the organizational culture adopted by the company?

How many resources are designated for the project?
What is the process used by the company?

---

**D.2 QT2 – RiPLE-SC Analysis Questionnaire**

Date:
Name:
Degree: ( ) Graduate  ( ) M.Sc.  ( ) Ph.D  ( ) Specialization

How many software projects did you participate after graduate?

**Academic**

<table>
<thead>
<tr>
<th>Category</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPL</td>
<td>Other Projects</td>
</tr>
<tr>
<td>SPL</td>
<td>Other Projects</td>
</tr>
<tr>
<td>SPL</td>
<td>Other Projects</td>
</tr>
</tbody>
</table>

- Low complexity:
- Medium complexity:
- High complexity:

**Industrial**

<table>
<thead>
<tr>
<th>Category</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPL</td>
<td>Other Projects</td>
</tr>
<tr>
<td>SPL</td>
<td>Other Projects</td>
</tr>
<tr>
<td>SPL</td>
<td>Other Projects</td>
</tr>
</tbody>
</table>

- Low complexity:
- Medium complexity:
- High complexity:

Please, inform which courses or disciplines you attended related to SPL.
What are your experience in scope definition, i.e., your experience to identify features, define scope, and domain analysis?

**Academic**

Identify features
- Low
- medium
- High

Define features scoping
- Low
- medium
- High

Domain analysis
- Low
- medium
- High

**Industrial**

Identify features
- Low
- medium
- High

Define features scoping
- Low
- medium
- High

Domain analysis
- Low
- medium
- High
Which are the techniques or practices that you know to scope definition?
_______________________________________________________
_______________________________________________________
_______________________________________________________

Did you have some difficult to apply or understand the scoping process?
_______________________________________________________
_______________________________________________________
_______________________________________________________

In your opinion, which are the strengths of the RIPE-SC process?
_______________________________________________________
_______________________________________________________
_______________________________________________________

In your opinion, which are the drawbacks of the RIPE-SC process?
_______________________________________________________
_______________________________________________________

Is there any missing task, role or artifact in the process?
_______________________________________________________
_______________________________________________________

Does the process aid in the execution agility of the tasks, i.e., it is effective in the purpose to be an agile process?

☑ Excellent ☐ Good ☐ Inferior ☐ Terrible
Justify:
_______________________________________________________
_______________________________________________________
_______________________________________________________

Which improvements would you suggest for the RiPLE-SC process?
_______________________________________________________
_______________________________________________________
_______________________________________________________
What is your satisfaction using the RiPLE-SC process?

☐ Very satisfied ☐ Satisfied ☐ Unsatisfied ☐ Very unsatisfied

Justify:

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

Did you have sufficient orientation in the use of the process?

☐ Most of the time ☐ Sometimes ☐ Rarely ☐ Never

Justify:

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
Appendix E. Meeting Minutes

In this appendix all the meeting minutes occurred during the execution of the RiPLE-SC are described.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Pre-Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Pre-Scoping Meeting</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Author of this dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Customer and Domain Expert (Project Manager and the Business Analyst from MedicWare)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>1h:19min</td>
</tr>
<tr>
<td>Data</td>
<td>January 04, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Pre-Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Analyze Market</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Author of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert and Domain Expert (Market Expert from MedicWare)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>1h:56min</td>
</tr>
<tr>
<td>Data</td>
<td>January 07, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
<tr>
<td>Phase</td>
<td>Domain Scoping</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Additional Task</td>
<td>Documentation Evaluation</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Author of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RISE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RISE)</td>
</tr>
<tr>
<td></td>
<td>Developer (RISE)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:33min</td>
</tr>
<tr>
<td>Data</td>
<td>January 05, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>Federal University of Bahia</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Domain Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Analyze domains</td>
</tr>
<tr>
<td>Steps</td>
<td>Review Domains and Identify Sub-Domains</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Author of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RISE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RISE)</td>
</tr>
<tr>
<td></td>
<td>Developer (RISE)</td>
</tr>
<tr>
<td></td>
<td>Customer and Domain Expert (Project Manager and the Business Analyst)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:54min</td>
</tr>
<tr>
<td>Data</td>
<td>January 06, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
<tr>
<td>Phase</td>
<td>Domain Scoping</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Task</td>
<td>Analyze domains</td>
</tr>
<tr>
<td>Step</td>
<td>Analyze Sub-Domains</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Author of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Scoping Expert (Project Manager from RiSE – Co-Advisor of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Customer and Domain Expert (Project Manager and the Business Analyst)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:30min</td>
</tr>
<tr>
<td>Data</td>
<td>January 08, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Domain Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Analyze domains</td>
</tr>
<tr>
<td>Step</td>
<td>Analyze Sub-Domains</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Project Manager from RiSE – Co-Advisor of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Customer and Domain Expert (Project Manager)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:31min</td>
</tr>
<tr>
<td>Data</td>
<td>January 13, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
<tr>
<td>Phase</td>
<td>Product Scoping</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Task</td>
<td>Identify Features – Workshop 1</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Project Manager from RiSE – Co-Advisor of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:56min</td>
</tr>
<tr>
<td>Data</td>
<td>January 20, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>Federal University of Bahia</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Identify Features - Workshop 2</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Project Manager from RiSE – Co-Advisor of this Thesis)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>3h:40min</td>
</tr>
<tr>
<td>Data</td>
<td>January 21, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>Federal University of Bahia</td>
</tr>
<tr>
<td>Phase</td>
<td>Product Scoping</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Task</td>
<td>Identify Features - Workshop 3</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Project Manager from RiSE – Co-Advisor of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:48min</td>
</tr>
<tr>
<td>Data</td>
<td>January 25, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>Federal University of Bahia</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Identify Features – Workshop 4</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Project Manager from RiSE – Co-Advisor of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:58min</td>
</tr>
<tr>
<td>Data</td>
<td>January 29, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>Federal University of Bahia</td>
</tr>
<tr>
<td>Phase</td>
<td>Product Scoping</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Task</td>
<td>Identify Features – Workshop 5</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Project Manager from RiSE – Co-Advisor of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:40min</td>
</tr>
<tr>
<td>Data</td>
<td>February 04, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>Federal University of Bahia</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Identify Features – Workshop 6</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Project Manager from RiSE – Co-Advisor of this Dissertation)</td>
</tr>
<tr>
<td></td>
<td>Product Line Managers (Project Manager and the Project Leader from RiSE)</td>
</tr>
<tr>
<td></td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Architect (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:46min</td>
</tr>
<tr>
<td>Data</td>
<td>February 10, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>Federal University of Bahia</td>
</tr>
<tr>
<td>Phase</td>
<td>Product Scoping</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>Task</td>
<td>Identify Features – Training 1</td>
</tr>
<tr>
<td>Participants</td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (MedicWare)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:17min</td>
</tr>
<tr>
<td>Data</td>
<td>February 25, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Identify Features – Training 2</td>
</tr>
<tr>
<td>Participants</td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (MedicWare)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:12min</td>
</tr>
<tr>
<td>Data</td>
<td>February 26, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Identify Features – Training 3</td>
</tr>
<tr>
<td>Participants</td>
<td>Two Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (MedicWare)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:17min</td>
</tr>
<tr>
<td>Data</td>
<td>March 01, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
<tr>
<td>Phase</td>
<td>Product Scoping</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Task</td>
<td>Features Review Meeting – first meeting</td>
</tr>
<tr>
<td>Participants</td>
<td>Three Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (MedicWare)</td>
</tr>
<tr>
<td>Duration</td>
<td>3h:12min</td>
</tr>
<tr>
<td>Data</td>
<td>January 20, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Features Review Meeting – second meeting</td>
</tr>
<tr>
<td>Participants</td>
<td>Three Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (MedicWare)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:00min</td>
</tr>
<tr>
<td>Data</td>
<td>January 25, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Features Review Meeting – third meeting</td>
</tr>
<tr>
<td>Participants</td>
<td>Three Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (MedicWare)</td>
</tr>
<tr>
<td>Duration</td>
<td>1h:30min</td>
</tr>
<tr>
<td>Data</td>
<td>February 04, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
<tr>
<td>Phase</td>
<td>Product Scoping</td>
</tr>
<tr>
<td>-------</td>
<td>----------------</td>
</tr>
<tr>
<td>Task</td>
<td>Features Review Meeting – fourth meeting</td>
</tr>
<tr>
<td>Participants</td>
<td>Three Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (MedicWare)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:00min</td>
</tr>
<tr>
<td>Data</td>
<td>February 26, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Features Review Meeting – fifth meeting</td>
</tr>
<tr>
<td>Participants</td>
<td>Three Developers (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (RiSE)</td>
</tr>
<tr>
<td></td>
<td>Developer (MedicWare)</td>
</tr>
<tr>
<td>Duration</td>
<td>1h:18min</td>
</tr>
<tr>
<td>Data</td>
<td>March 02, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Product Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Construct product Map</td>
</tr>
<tr>
<td>Participant</td>
<td>Scoping Expert</td>
</tr>
<tr>
<td>Duration</td>
<td>7h:18min</td>
</tr>
<tr>
<td>Data</td>
<td>February 03, 2010 to March 22, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
<tr>
<td>Phase</td>
<td>Assets Scoping</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Task</td>
<td>Validate Product Map</td>
</tr>
<tr>
<td>Participants</td>
<td>Manager (MedicWare)</td>
</tr>
<tr>
<td></td>
<td>Market Expert (MedicWare)</td>
</tr>
<tr>
<td></td>
<td>Three Developers (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:3min</td>
</tr>
<tr>
<td>Data</td>
<td>March 26, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Assets Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Define Metrics</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Author of this Dissertation)</td>
</tr>
<tr>
<td>Duration</td>
<td>19h:45min</td>
</tr>
<tr>
<td>Data</td>
<td>January 19, 2010 to January 25, 2010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Assets Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Apply Metrics</td>
</tr>
<tr>
<td>Participants</td>
<td>Scoping Expert (Author of this Dissertation)</td>
</tr>
<tr>
<td>Duration</td>
<td>45h:00min</td>
</tr>
<tr>
<td>Data</td>
<td>March 30, 2010 to April 15, 2010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Assets Scoping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Prioritize Product Map</td>
</tr>
<tr>
<td>Participants</td>
<td>Manager (MedicWare)</td>
</tr>
<tr>
<td></td>
<td>Product Line Manager</td>
</tr>
<tr>
<td></td>
<td>Three Developers (RiSE)</td>
</tr>
<tr>
<td>Duration</td>
<td>2h:00min</td>
</tr>
<tr>
<td>Data</td>
<td>April 22, 2010</td>
</tr>
<tr>
<td>Place</td>
<td>MedicWare</td>
</tr>
</tbody>
</table>