SCOPE DETERMINATION IN SOFTWARE PROCESS LINES

PABLO HERNANDO RUIZ MELENJE

Tesis de Doctorado en Ciencias de la Electrónica

Director:
Julio Ariel Hurtado Alegría
PhD. en Ciencias, Mención Computación

Facultad de Ingeniería Electrónica y Telecomunicaciones
Doctorado en Ciencias de la Electrónica
Grupo de Investigación y Desarrollo en Ingeniería de Sistemas (IDIS)
Áreas de Investigación: Ingeniería de Software
Popayán, septiembre de 2019

PABLO HERNANDO RUIZ MELENJE

SCOPE DETERMINATION IN SOFTWARE PROCESS LINES

Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la Universidad
del Cauca para la obtención del título de Doctor en
Ciencias de la Electrónica

Doctor en:
Ciencias de la Electrónica

Director:
Julio Ariel Hurtado Alegría
PhD. en Ciencias, Mención Computación

Popayán
2019

I would like to dedicate this thesis to my whole family. Especially to my daughter, Maria Paula, for being the love of my life and my everything. To my parents Amadeo and Flor, for giving me life, advice, and education. To my brothers Dario and Jimena for their complicity and support. All were an essential part of this achievement.

Acknowledgment

To my family, for their unconditional support. I especially thank my parents Flor and Amadeo, for their example, for their love, for their time, perseverance and strength, for being a source of inspiration and tenacity.

To Julio Ariel, my director and friend, who has been the primary manager of this whole process, who has guided and unconditionally supported me to achieve this great goal.

To my girlfriend and friend Vanessa Agredo, who supported me and encouraged me to continue, when it seemed that I was going to give up.

To the University of Cauca, to the Doctorate Committee, to the InnovAccion project, to Unicomfacauca for the financial support provided that made this study possible.

Resumen

El proceso de software es una forma reconocida de guiar el desarrollo de software, sin embargo, tener un proceso no significa necesariamente que se adapte a todas las situaciones específicas. En consecuencia, los procesos de software normalmente requieren ser adaptados a cada contexto específico para lograr el proceso adaptado más adecuado. Sin embargo, la adaptación del proceso de software es un área de investigación desafiante porque es una actividad intensiva que requiere conocimiento: requiere experiencia, conocimiento y requiere mucho tiempo. En los últimos años, han surgido diferentes enfoques para adaptar el proceso del software. Una de las estrategias recientes para adaptar los procesos de software es planificar y administrar su variabilidad a través de una estrategia de producción siguiendo un enfoque de Ingeniería de líneas de proceso de software (SPrLE, por sus siglas en ingles), que se centra en definir una línea de proceso de software (SPrL, por sus siglas en ingles) en lugar de procesos independientes. En la Ingeniería de líneas de proceso de software, una actividad crucial es la delimitación del alcance de los SPrL, donde se establecen los procesos y las características de los procesos que se reutilizan, así como también se especifican las situaciones en las que estos procesos serán adecuados. La definición del alcance es una actividad compleja y crítica, que puede facilitar u obstaculizar a las organizaciones para lograr una solución SPrL viable. Además, el enfoque de definición del alcance no tiene una dirección clara para que la industria del software pueda hacer su incorporación de manera apropiada. La falta de elementos de guía en la extracción, análisis y modelado del alcance hace que estos enfoques sean inapropiados para dirigir esta actividad vital cuya definición impacta en el éxito de la definición de líneas de proceso de software. Esta tesis presenta el enfoque SpeTion-SPrL (ScoPE determinaTION in Software Process Lines, por sus siglas en ingles) para la determinación sistemática del alcance en las líneas de proceso de software. Se basa en la identificación de la necesidad de proyectos y productos con respecto a los elementos del proceso. Identifica las necesidades y su correspondiente correlación con los activos del proceso a través de un nivel de idoneidad que permite seleccionar cuantitativamente y tomar decisiones de adaptación sobre los elementos del proceso. SpeTion-SPrL se construyó bajo los principios de sistematicidad, integralidad, adecuadamente conducido y materializable, cuyo soporte está determinado por los diferentes elementos que lo componen. Integra elementos de tres enfoques principales: alcance SCOPE, CASPER y SPL, se definió utilizando técnicas sistemáticas y se perfeccionó a través de su evaluación.

Palabras Clave: Líneas de procesos de software, definición del alcance en líneas de procesos de software, procesos, familias de procesos**.**

Abstract

Software process is a recognized way for guiding the software development, however having a process does not necessarily mean, that it fits all specific situations. Consequently, the software processes normally require being tailored to each specific context to achieve the rightest adapted process. However, software process tailoring is a challenging research area because it is a knowledge-demanding intensive activity: it requires experience, knowledge and it is time-consuming. In recent years, different approaches have emerged for tailoring the software process. One of the recent strategies for tailoring software processes is to plan and manage their variability through a production strategy following a Software Process Lines Engineering (SPrLE) approach, which focuses on defining a software process line (SPrL) instead of independent processes. In Software Process Line Engineering (SPrLE) a crucial activity is the delimitation of the scope of the SPrLs, where establishing the processes and the characteristics of the processes that reuse, as well as specify the situations in which these processes they will be suitable. The scope definition is a complex and critical activity, which may facilitate or hinder the organizations from achieving a viable SPrL solution. Furthermore, the scope definition approach does not have a clear direction so that the software industry can make its incorporation in an appropriate way. The lack of guiding elements in the extraction, analysis, and modeling of the scope makes these approaches inappropriate for direction this vital activity whose definition impact in the success of the software process lines definition. This thesis presents SpeTion-SPrL (ScoPE determinaTION in Software Process Lines) approach for systematic scope determination in the Software Process Lines. It is based on the identification of the need for projects and products with respect to the process elements. It identifies the needs and their corresponding correlation with the process assets through a suitability level that allows quantitatively select and make adaptation decisions on the process elements. SpeTion-SPrL was built under the principles of systematicity, integrality, properly conducted and materializable, whose support is determined by the different elements that make it up. It integrates elements of three main approaches: SCOPE, CASPER and SPL scoping and it was defined using systematic techniques and refined through its evaluation.

Keywords: Software Process Lines, Scoping in software process Lines, process, Software process families.

Content

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

This chapter gives a thesis motivation that comes from three aspects of research in software engineering, such as the software process, software process reuse, and software Process tailoring. Then, the main problems that the Software Process Lines have for their construction and definition are presented and the problem statement is introduced. Further, the objectives and the hypothesis are presented, as well as the research method followed for thesis advance, and finally, the main research contributions and the thesis structure are summarized.

1.2 Motivation

Despite the growing interest of the research community on software processes (Carvalho & Chagas, 2014) there are still open topics on how to provide the software industry with practical approaches according to the real needs to do the management, reuse, and adaptation of the software processes and their assets (Costa, Nogueira Teixeira, & Lima Werner, 2018), (Barreto, Murta, & Cavalcanti da Rocha, 2011). The above is the incentive of this thesis wherein the following sections is shown in greater detail each of the elements that are part of our motivation.

1.2.1 Software Process

A significant part of software projects reveals problems at achieving delivery time of products covering the user expectations (Standish Group, 2012). Most of the identified causes are due to the absence of disciplined approaches in software development. A disciplined software process increases productivity and the quality of software development, which is why they are considered as one of the best mechanisms to manage and control projects in the construction of software products (Xu, 2005). There is a significant number of life cycle and software processes models available from industry and literature, including reference models such as CMMI (SEI, 2006), ISO/IEC 12207 (ISO, 2008); prescriptive software process models such as the Unified Process Model (Jacobson, Booch, & Rumbaugh, 1999), and agile approaches such as XP (Beck & Andres, 2004) and Scrum (Schwaber, 1997). Many of the problems in software development are due to the lack of disciplined approaches to management the projects due to the organizations do not adequately manage and adapt the processes and their assets.

1.2.2 Software process reuse

One strategy to accelerate process improvement is to replicate standard organizational processes within other projects (Barreto, Murta, & Cavalcanti da Rocha, 2011). However, the creation of a process that is usable in various projects is a difficult task (Ocampo, Bella, & Münch, 2005), (Ruiz, Camacho, & Hurtado, 2018). What is needed is an effective method to capture the common and variable elements of the project processes specific and produce processes definitions that can be applied in a variety of situations, that is, that are reusable (Barreto, Murta, & Cavalcanti da Rocha, 2011). A reusable process can be defined as the use of a description of a process in the creation of another process (Hollenbach & Frakes, 1996). To achieve reuse in software processes, different concepts of the reuse of software products have been adapted as components, architectures, product lines, and reuse patterns. (Barreto, Murta, & Cavalcanti da Rocha, 2011). This adaptation has been made in order to transfer the benefits of the reuse of software products to the software processes such as the reduction of re-work, increase productivity, improve quality, decrease cost/effort, decrease the time required to perform the activities. However, there are also some difficulties reusing in software products, which also affect the reuse of processes such as difficulty in identifying, retrieving and modifying reusable elements, insufficient quality of reusable elements, the need to create initiatives of reuse, high adoption cost, and inadequate support tools. (Barreto, Murta, & Cavalcanti da Rocha, 2011).

1.2.3 Software Process tailoring

Software companies typically do not adapt their standard processes to their specific project’s needs; this is mainly due to the complexity and knowledge required for this adaptation (Pedreira, Piattini, Luaces, & Brisaboa, 2007). Normally, what companies do is adapt their specific projects to the standard process, this has negative consequences for the projects since working with the inadequate process can generate more complexity, costs and development times by using an over-sizing process to the needs of the project. Additionally, if the process is tailored in an unsystematic way, it is possible to exclude essential elements of process that are required for a good project execution, using an under-sizing process (Ruiz, Camacho, & Hurtado, 2018). Software process tailoring is "the act of adjusting the definition and/or particularizing the terms of a general description to derive a description applicable to an alternate (less general)" (Ginsberg & Quinn, 1994). The software process should be tailoring to satisfy characteristics both enterprises like software projects (Armbrust, Ebel, Hammerschal, Münch, & Thoma, 2008), (Jacobson, Booch, & Rumbaugh, 1999), (Humphrey, 1989). Literature has identified the software process tailoring as a hard activity, because it requires experience, it is knowledge intensive and time-consuming (Ocampo, Bella, & Münch, 2005), (Hurtado & Bastarrica, 2012). Software process tailoring is a challenging research area; particularly, there are several unsolved problems such as: how an adaptable software processes could be defined? how process assets could be managed? what techniques, methods, practices and tools are appropriated for achieving a practical process tailoring? (Ruiz, Camacho, & Hurtado, 2018), (Hurtado & Bastarrica, 2012). Therefore, the software process tailoring still requires careful and hard work (Hanssen, Westerheim , & Bjørnson, 2005).

1.3 Problem statement

One of the approaches that provide a systematic and repeatable way in the adaptation, reuse and management of the software process are the Software Process Lines (SPrL), (Ruiz & Hurtado, 2016), (Hurtado & Bastarrica, 2012) , (Barreto, Murta, & Cavalcanti da Rocha, 2011) which allow establishing a reference process from which others processes could be derivate (Rombach, 2005). According to the state of practice and of the literature in SPrL have been identified the following problems, which in this thesis will are addressed through our approach proposal.

1.3.1 Immature empirical evidence

Since it was considered that the concept and the benefits provided by the reuse in product lines could be applied in a similar way in the processes field (Rombach, 2005), specifically in the software process lines, its research has increased in recent years (Carvalho & Chagas, 2014). However, the different proposals raise approaches that do not provide adequate support in real contexts and show a lower level of maturity with respect to the business processes lines and SPL (Schramm, Dohrmann, &

Kuhrmann, 2015). The reports growing number of experiences in industry and academia shows that the SPrL approach is feasible and beneficial to be applied in real contexts. However, it is still considered an immature area with many open topics, such as: *a)* Lack of modeling of standards and well-known process models that use SPrL concepts. *b)* Need to improve the approaches evaluation quality in terms of empirical validation (Carvalho & Chagas, 2014). Approach (Schramm, Dohrmann, & Kuhrmann, 2015). *e)* There are no comparative studies to highlight the advantages and disadvantages of the proposed approaches (Carvalho & Chagas, 2014). In this way, we appeal the request to do more research on SPrL concepts applicability in practice (Carvalho & Chagas, 2014) , (Chen, Babar, & Ali, 2009), (Blum, Simmonds, & Bastarrica, 2015), (Schramm, Dohrmann, & Kuhrmann, 2015), to provide empirical evidence and support in achieving the maturity of this promising area and benefit the companies with the proper management, reuse, and adaptation of their software processes (Schramm, Dohrmann, & Kuhrmann, 2015).

1.3.2 Scope definition is a key and complex activity that influence the successful of SPrL approach

A key activity for building a SPrL, is the SPrL scope determination, because it enables to software organizations to achieve a cost-effective SPrL solution (Ruiz & Hurtado, 2016). Armbrust et al. (2009) stated that *“a software process line scoping is the identification of the range of characteristics that processes in the process line should cover. SPrL scoping will determine situations where the process will be used and what process elements (common and variables) will be required in each situation*”. Consequently, the scope definition is a complex task because it requires mix complex information related to processes, process assets, process features, with previous experiences in software process adaptation and potential new situations. In the same way, the scope definition must provide the necessary process elements to define the appropriate processes supporting the creation of current and future products and projects. Therefore, it should provide a systematic method for classifying process parts accordingly, so that process management is facilitated. Furthermore, some relevant questions that process engineers must address are (Ruiz, Camacho, & Hurtado, 2018): which processes will be part of the SPrL? What current and future processes make part of it? Under what criteria the process engineer includes processes in a SPrL? How to manage/evolve the scope of SPrL? Under what criteria variability could be added or excluded in a SPrL? How SPrL scope should be guide? According to the last, the SPrL scope definition comprises solve some challenges that difficult organizations to achieve a feasible SPrL solution.

1.3.3 SPrL Scope definition does not have a clear direction

The importance of scoping is already well recognized in the area of software reuse (Schmid & Gacek, 2000), (Armbrust, y otros, 2009). However, the approaches sometimes forget the importance and necessity of adequate scoping definition in a systematic way (Schmid K. , 2000), (Armbrust, y otros, 2009), (Carvalho & Chagas, 2014). Therefore, the relevant approaches and studies (Armbrust, y otros, 2009), (Armbrust, y otros, 2008), (Hurtado & Bastarrica, 2012) for scope definition in software process lines do not have a clear direction so that the software industry can make its incorporation in an appropriate way. In the definition of the scope, there is no clarity about how the processes, the process assets, and the specific variabilities will be part of the reuse infrastructure, as well as at what stage of construction of the SPrL the scope will have to be materialized (Ruiz, Camacho, & Hurtado, 2018), moreover, the amount of refinement is not known that this activity requires to achieve an adequate definition. The lack of guiding elements in the extraction, analysis, and modeling of the scope makes these approaches inappropriate for direction this vital activity whose definition impact in the success of the software process lines definition. Therefore, the scope does not have clear guidelines in its definition, which is why it can be considered ambiguous and difficult to define (Ruiz, Camacho, & Hurtado, 2018).

1.3.4 SPrL variability and tailoring strategy depends on scope determination

In SPrLs scoping definition the situations and the process variability, with its possible relationships, are identified and a strategy for generating processes is selected/created (Armbrust, y otros, 2008), (Ruiz, Camacho, & Hurtado, 2018). . The identification of different situations allows to process engineer defining in a planned way a set of current and future needs that the processes that the SPrL must support. The variability identification aims to provide flexibility and adaptability to deal with the situations that SPrL will face. Correctly defining process variability is crucial, due to that a narrow variability does not allow adequate exploitation of reuse opportunities and a wide variability will involve the development of the assets on which much effort is wasted, therefore, having a balance of the variability is vital for the SPrL to provide process solutions according to specific situations. Furthermore, for scoping, it is necessary to establish the relationships between the situations and the variability to support the adaptation rules guiding the production strategy execution. Hence, a good definition of the scope must adequately provide all the aforementioned elements in a way that maximize the results of the SPrL implementation effort.

1.4 Research Goals

1.4.1 Overall goal

To establish a scope determination mechanism which helps in its extraction and specification during the SPrL construction.

1.4.2 Specific goals

- To characterize, according to the literature, mechanisms such as methods, techniques, practices, tools, process and strategies for determining the scope in both SPrL and SPL solutions.

- To define, adapt, refine and simplify the necessary elements that will be part of a scope determination mechanism in order that supports the SPrL approach incorporation in industrial contexts.

- To evaluate the appropriateness[1] of the mechanism using the case study method.

1.5 Hypothesis

For evaluating the goals described above the following hypothesis was defined:

An integrated scope determination mechanism systematically guides to process stakeholders for gathering a well specified and appropriate software process line scope in both experimental and industrial settings.

Figure 1.**1** shows the overall relationships between the motivation, the problem statement, goals and the hypotheses derived by this thesis.

---

[1] Appropriateness in this work make reference to: That meets the necessary elements to guide definition and extraction of the scope

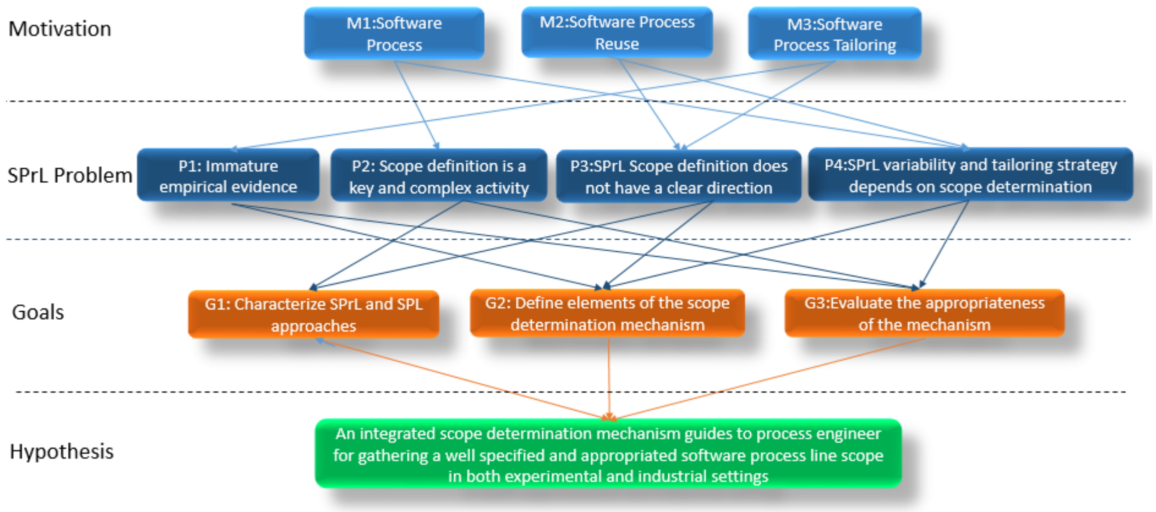


Figure 1.1. Relationships between the motivation, the problem statement, goals and the hypotheses

1.6 Research approach

In order to achieve the project objectives, the scientific method described by Bunge (Philosophy of Science, 2002) was selected as the empirical research framework, which was executed in an iterative and incremental way. Moreover, the case study research method proposed by Yin (Yin, 1994) with the guidelines propose by (Runeson & Höst, 2009) will be used in order to explore and confirm the project hypothesis. The complete research method was adapted used three points of view. The first one refers to a management view, the second one refers to the research disciplines, main activities proposals by Bunge (Philosophy of Science, 2002), and third one refers to the research iteration.

Figure 1.**2**. shown the three phases and their milestones are: exploration (problem statement), formulation (approach) and validation (hypothesis validated).

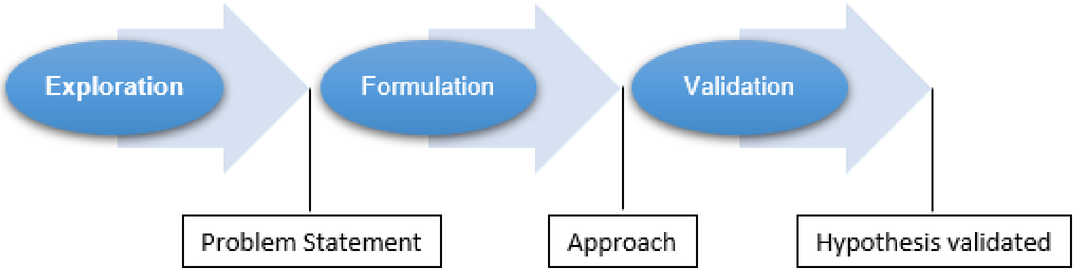


Figure 1.2. Phases and milestones of Research method

The three phases and their milestones are: exploration (problem statement and hypothesis), formulation (hypothesis refinement) and validation (Thesis). In the exploration phase, the project proposal was achieved (Thesis I), by the end of the formulation phase our approach was defined and when the validation was performed, the research was culminated (Thesis II).

The second point view refers to the research disciplines: Research problem definition, theoretical model construction, specific consequences deduction, proof of hypothesis, Introduction of conclusions to the theory and documentation. These disciplines were executed many times with different levels of effort according to the phase in execution as it is shown in Figure 1.**3**.

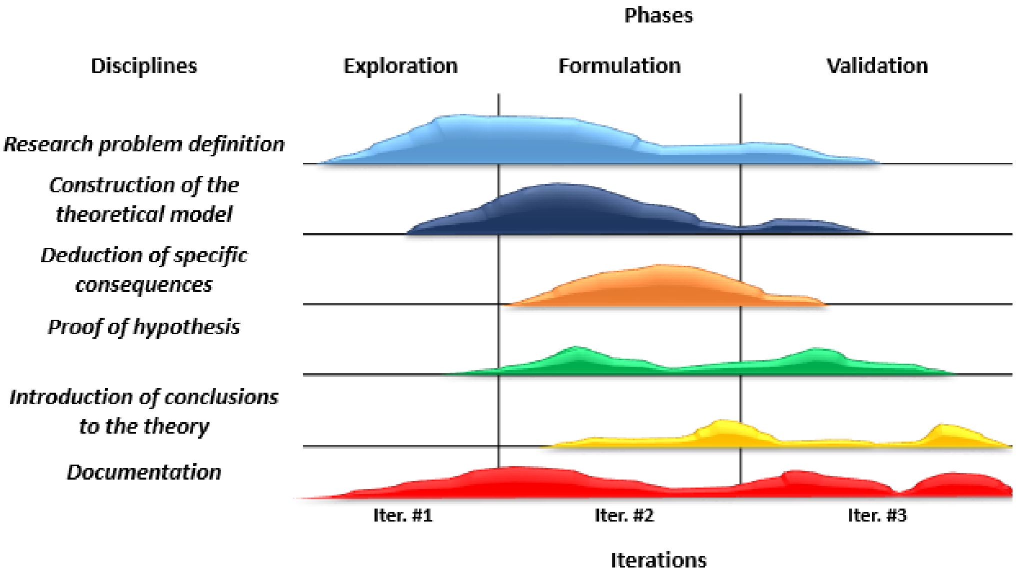


Figure 1.3. Research Disciplines

Subsequent activities for the development of the project are as follows:

1. *Research problem definition:*

Facts recognition: facts revision, preliminary classification and selection of those that are most likely relevant.

- Problem discovering: finding a gap or missing elements in the SPrL literature. Studying theoretical references about SPrL approaches such as methods, frameworks, techniques and referenced experiences.
- Problem formulation: problem statement establishing via research question, i.e. to reduce the problem to its essential core through a comprehension of the available knowledge.

2. *Construction of the theoretical model*:

- Selection of relevant factors: a study of the different software process scoping definition approaches in the SPrL and SPL context was realized, in order to achieve a structured characterization of the mechanisms and strategies for scoping of software processes.
- Stablishing of the central hypotheses and the auxiliary assumptions: it include to propose a set of assumptions related to the connections between the relevant variables.
- Construction of the scope determination mechanism: For the development of this activity different mechanisms and strategies were considered which will be part of the mechanism proposed
- Define a proof of concept: the mechanisms proposed was used by performing a proof of concept to verify its initial behavior via exploratory case study.

3. *Deduction of specific consequences*:

- Search for empirical support: elaboration of predictions based on a theoretical model and empirical data gathered exploratory case studies and specific experiences. Predictions preparation or expected results for the SPrL scope through the proposed mechanism.

4. *Proof of hypothesis*

- Case planning: planning the cases study in order to proof the SPL scope determination mechanisms in a software company.
- Case selection: selection and preparation of the case study according to the research objectives.
- Case study design: this activity will be conducted by the case study method proposed by (Runeson & Höst, 2009).
- Execution of the case: performance of the operations and data collection through case study. In this activity a software process line scoping will be defined through an uncontrolled environment (real phenomena), using the mechanism proposed in a real software company.
- Proof conclusion: interpretation of the gathered data from the case study point of view.

5. *Introduction of conclusions to the theory:*

- Confrontation of the results obtained with the predictions
- Relevant adjustments to the method: analysis of results and necessary adjustments to the mechanism
- Suggestions about further work: search for gaps or errors in theory and / or empirical procedures, if the model has been disconfirmed; if confirmed, examination of possible extensions and possible consequences.

6 Documentation

- Writing scientific papers, technical reports, process models
- Writing of the thesis document

The above two points view, are related in a new third point view: the research iteration. A research iteration organizes the research disciplines in a workflow as it is presented in the Figure 1.3. In each iteration, the variable emphasis by phase is represented in Figure 1.4. In this thesis three iterations were executed with the following results:

- First iteration: A Process Line was defined based on the Unified Processes in which an initial version of the problem and the state of the technique were evidenced and established. The hypothesis was initially stated. Two publications were obtained: A Software Process Line Based on the Unified Process", (Ruiz & Hurtado, 2013) and

Variability analyzing of the Unified Process with the Washizaki Technique (Ruiz, Hurtado, & Camacho, 2014). See *Annex A and B*.

- Second Iteration: a more complete description of the problem and the state of the art was established. The hypotheses were stated with a better definition than in the previous iteration. An empirical study was defined in GreenSQA company in order to know how some approaches to scope definition works in a real context. The first scope approach version was established; this proposal was validated through experimentation in a university context. In this iteration six articles were made: Framework Based on Software Process Lines for Tailoring of Software Process Models (Ruiz & Hurtado, 2016). A Comparative Study for Scoping a Software Process Line (Ruiz, Camacho, & Hurtado, 2018), Customizable Software Processes: An Exploratory Case Study in a Small Organization (Ramírez, Lasso, Ruiz, & Hurtado, 2018), A canonical software process family based on the Unified Process (Ruiz, Camacho, & Hurtado, 2018), (Camacho, Alvarez, Hurtado, & Ruiz, 2019), (Ruiz & Hurtado, 2017) and (Ruiz P. , Camacho, Marques, Hurtado, & Bastarrica, 2019).See annexes*, C, D, F,* G, *H, I* respectively.

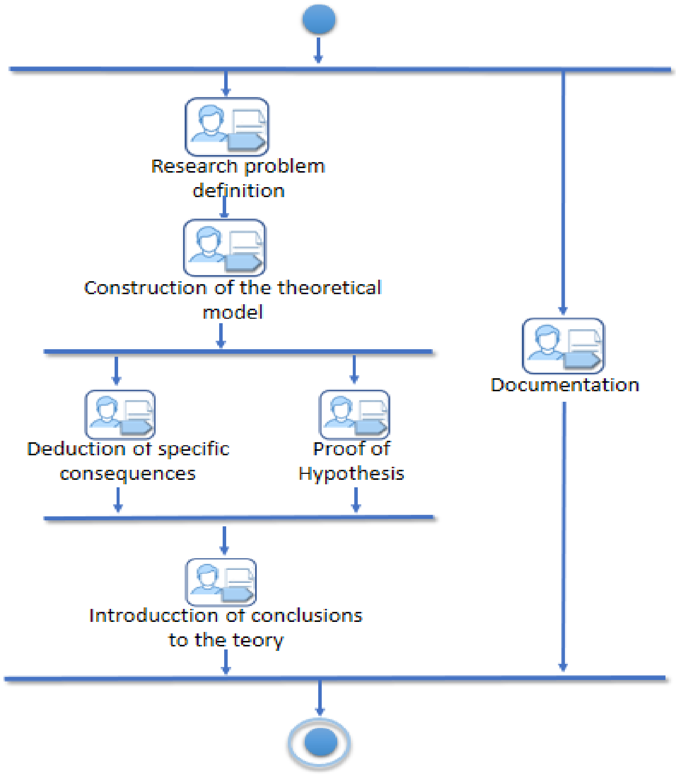


Figure 1.4. Research method Iteration

- Third Iteration: industrial cases were defined to validate the hypothesis. Mechanisms were improved and applied in the study case. In this iteration two articles were made: An experiment in scoping definition in software process lines (Ruiz, Agredo, Mon, & Hurtado, 2019), An Empirical Study for scoping a Software Process Line for Testing Services (Ruiz & Hurtado, 2019).

1.7 Research contributions

According to (Carvalho & Chagas, 2014), this research aims to contribute to the definition of a scope determination approach in Software Processes Lines in such a way as to allow facilitate the incorporation of this approach in industrial settings. Particularly the contributions of the research are:

- To define and structure a state of the art in which the importance of the scope can be evidenced, as well as the need to improve and contribute to the development of empirical evidence in the use of SPrL
- To make evident the scope importance, as well as the need to improve and contribute to the development of empirical evidence in the use of the SPrL approach through the definition and structuration of the state of the art. It was also evident little systematicity in the scope definition with the CASPER approach and little granularity of the SCOPE method as part of the result of an empirical study
- To contribute to the adoption of software process lines by defining a systematic approach to scope determination. SpeTion-SPrL attempts to provide a series of strategies to guide the software industry in defining the scope in an appropriate way, which is one of the crucial activities for the successful implementation of the SPrL approach. SpeTion-SPrL was built under the principles of systematicity, integrality, properly conducted and materializable, whose support is determined by the different elements that make it up.
- To provide the community with empirical evidence on the application of software process lines and the scope concept

1.8 Thesis structures

The rest of this document contains the following sections: Chapter 2 includes the state of the art and the related work about software product and process lines, and software and process lines scoping. Chapter 3 details the SpeTion- SPrL approach including the elements that compose it. Chapter 4 report the SpeTion-SPrL application in experiments and case studies. Chapter 5 presents the conclusions and some further work.

Chapter 2

Background and state of art

2.1 Overview

This chapter provides a contextualization of some important concepts, definitions and advances involved in this thesis. Moreover, it gives an overview of some main approaches categories in software process tailoring, SPrL approaches, techniques in SPrL approaches, SPrL scoping and SPL scoping.

2.2 Background

2.2.1 Software Process

A software process is the set of tools, methods, and practices for producing a software artifact (Humphrey, 1989), (Hossein & Natsu, 1997), (Ginsberg & Quinn, 1994). For Ginsberg et al. (Ginsberg & Quinn, 1994) Software process is defined as a set of activities, methods, practices and transformations that people use to develop and maintain software, as well as their associated products (e.g. plans, specifications, designs and testing). For Xu et al. (Xu, 2005) a software process is a set of activities necessary to transform user requirements into a software system. Similarly, Acuña et al. (2001) state that a software process is a partially ordered set of tasks undertaken to manage, develop and maintain software systems.

2.2.2 Software process modeling

Software process modelling describes the construction of software development process models. A software process model is a graphical or textual abstraction of the architecture, design or definition of the software process (Feiler & Humphrey, 1993). For Acuña et al. (Acuña & Ferré, 2001) the software process modelling refers to the definition of the processes as models, additionally any optional automated support available for modelling or executing the models during the software process.

2.2.3 Software product lines engineering and Software product lines

Software Product Line Engineering (SPLE) makes reference to a set of software engineering methods for building similar products from a reusable set of software assets following a common strategy for its production, the fundamental idea is to undertake the development of a set of products as a single, coherent development process (Schmid K. , 2000), which seeks to increase the software development companies productivity through of a planned construction of a software product set in order to meet a market segment needs which seeks to increase the software development companies productivity through of a planned construction of a software product set in order to meet a market segment needs (Northrop, y otros, 2007). According to Clements and Northrop (Clements & Northrop, 2001) a Software Product Line (SPL) is “a software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy 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”. SPL is an efficient and cost-effective approach to developing portfolios of similar products (Klaus, Günter, Linden, & Frank, 2005). Products are built from a core asset base, a collection of artefacts that have been designed specifically for use across the portfolio (Clements & Northrop, 2001), (Heradio-Gil R. , Fernandez-Amoros, Cerrada, & Cerrada, 2011). Some of the benefits pursued by the SPL are: improve the costs, schedule, and quality of software products through the assets reuse in a planned way (Clements P. , 2002).

2.2.4 Software process lines and Software process lines engineering

Rombach (Rombach, 2005) proposed that software process could be organized according to similarities and differences, allowing a better tailoring to specific project needs. Washizaki et al. (Washizaki, 2006) define a SPrL as “a set of processes in a particular domain or for a particular purpose, having common characteristics and built based upon common, reusable process assets”. For Armbrust et al. (Armbrust, y otros, 2009) a SPrL is “a set of software processes with a managed set of characteristics that satisfies the specific needs of a particular organization and that are developed from a common set of core processes in a prescribed way”. Moreover, for Ternité (Ternité, 2009) a SPrL is “a set of processes that captures commonalities and controlled variabilities “. Each of these processes is developed from a common set of core assets (features) in a prescribed way.

Process Lines Engineering (SPrLE) is the reference framework for building Software Process Lines (SPrL), it defines a process and necessary process components that must be during SPrL planning and materialization. SPrLE define two main processes: domain engineering (construction) and application engineering (adaptation and commissioning). Domain engineering refers to build the structural elements of a SPrL, and application engineering is responsible for deriving project-specific processes that satisfy specific situations (development of products and projects with similar characteristics) (Hurtado & Bastarrica, 2012) (Armbrust, y otros, 2008)

2.2.5 Software product line scoping

In the definition of product lines, a crucial activity is the definition of scope which looking for identify and bound the capacities (products, characteristics) and areas (subdomains, existing assets) where the investment in reuse is economically useful and beneficial for the development of the product (Schmid K. , Scoping software product lines, 2000), (John & Eisenbarth, 2009). In the scope definition activity, information is identified, captured and organized in order to make it reusable in the creation of new products belong to a specific domain. The scope captures the context, the most important requirements and the restrictions to derive the acceptance criteria for the final product. Therefore, it is clear the association of the scope with the success of the SPL, being necessary a very well defined scope process to efficiently implement product lines and reduce their risks (Moraes, Santana de Almeida, & Romero de Lemos Meira, 2009)

2.2.6 Software process line scoping

Armbrust et al. (Armbrust, y otros, 2008) proposed an approach of software process scoping. They define software process scoping as "*the systematic characterization of products, projects, and processes and the subsequent selection of processes and process elements, so that product development and project execution are supported efficiently, and process management effort is minimized*", but this approach is focused on process scope definition and it is not for software process lines. In later works Armbrust et al. (Armbrust, y otros, 2009) stated that a software process line scoping is the identification of the range of characteristics that processes in the process line should cover. SPrL scoping will determine situations where the process will be used and what process elements will be required in each situation. So, planned situations should be related to process features for determining suitable process configurations for each situation (Hurtado & Bastarrica, 2012).

2.3 State of art

2.3.1 Software process tailoring

2.3.1.1 Adjustment guides

The Capability Maturity Model Integration - CMMI recommends defining a library of process assets, have them available and define a general process that when applied would use a subset of assets. To determine that subset, the process must have an adjustment guide. The technical report of Ginsberg et al. (Ginsberg & Quinn, 1994) provides a framework and guide for process adaptation in organizations certified in the Capability Maturity Model for Software - CMM. The Unified Process (Jacobson, Booch, & Rumbaugh, 1999) is describe as a general framework for software development that provides an adjustment guide composed of recommendations to adapt the process to the specific needs of the project. The approaches above are an example of adaptation guidelines, but considering that, the adaptation is done manually, without previous planning, so they require great effort and expertise, which makes them prone to errors.

2.3.1.2 Process frameworks and asset reuse

There are framework-based adaptations, where generic process frameworks are defined and reused (Bustard & Keenan, 2005), (Lobsitz, 1996). Bustard et al. (Bustard

& Keenan, 2005) described a generic model to carry out the adaptation of the process; expressed in two frameworks: one whose objective is the collection and provision of information useful and necessary for the creation of the process, and an application framework, in which the process is designed, applied and evaluated. Lobsitz (Lobsitz, 1996) proposed a technical framework, which allows the selection of a specific architecture domain and a test and acceptance strategy; this framework also identifies an appropriate lifecycle model and the best deliverables and milestones for the process.

In the adaptation based on frameworks of process and reuse of assets it is established a general framework for processes assets organization to facilitate the reuse and adaptation of the process. In this kind of approach, a set of particularization mechanisms are provided to carry out the tailoring, such as the addition, elimination and configuration of the process elements (Yoon, Min, & Bae, 2001), (Fei & Tong, 2007). The tailoring final decisions fall on the process expert, which can be prone to error and consume too much time. In this same line of adaptation, there are also tools such as: EPFC[2] and RMC[3], to build process frameworks, which allow the selection of the set of process components that fit the specific needs of the project via process configurations. These are tools to help in the process adaptation, but they are not enough, because adaptation depends on the decisions made by the person in charge of adaptation and these decisions are no planned neither and repeatable.

2.3.1.3 Artificial intelligence

Some main approaches propose to use artificial intelligence for resolving the process tailoring: one based on neural networks (Xu, 2005), (Park, Naa, Parka, & Sugumaranb, 2006) and other based on CBR (Case Based Reasoning), (Henninger & Baumgarten, 2001), (Funk, 2000), (Dongwon, y otros, 2008), (Ahn, Ahn, & Park, 2003), (Lee & Lee, 2006). These approaches try to supply the expertise and experience of a process engineer to decide which and how adaptations would be made from previous data. Meanwhile Park et al. (Park, Naa, Parka, & Sugumaranb, 2006) proposed a semi-automatic adaptation method, in which neural networks based on learning theory are

---

[2] Eclipse Process Framework Composer (EPFC) es una un entorno de desarrollo gratuito para describir y visualizar proceso en SPEM.

[3] Rational Method Composer (RMC) es un entorno comercial de IBM para documentar y publicar procesos

used. The Park´s propose is adaptation method that only automates a single phase of the whole proposed method.

.Henniger et al. (Henninger & Baumgarten, 2001) used an organizational learning process and a CBR system based on rules and search for similarity to adapt process to the individual characteristics of the projects. CBR is also part of the method proposed by Funk (Funk, 2000), who identified and suggested the reuse of similar processes and tasks stored from a case repository. Dongwon et al. (Dongwon, y otros, 2008) performed a case recovery based on structural similarity, which is calculated by the portion of process elements from a previous case which are applicable to a new project. Meanwhile Ahn et. al. (Ahn, Ahn, & Park, 2003), used a hybrid method that combines CBR to facilitate the reuse of experiences, and knowledge-based techniques to derive process components. Lee et al. (Lee & Lee, 2006) developed a framework with the least modification principle - LMP for CBR, in order to assist the construction of the software project network. In general, CBR approaches were based on recovering the case of greater similarity with a new project, leaving aside the final adaptation of the recovered case that may be the most relevant of the entire process tailoring. Last modifications of the recovered case will influence decisively the fulfillment of the needs of the current project. For carrying out software process tailoring these approaches should consider the number of cases previously developed in a manual way (empirical evidence to derive the rules).

2.3.1.4 Situational method Engineering

Situational Method Engineering (SME) aims at providing techniques and tools that allow the construction of methods specific to each project (Ralyté, Deneckère, & Roll, 2003). This construction is developed through selecting process elements (fragments or pieces of the process) that have been created and stored in a repository or method base (Sellers & Ralyté, 2010). Search mechanisms and process element assembly are defined for the construction. Although search results yield related and coherent elements, this approach requires expertise and resources to build the specific process at the start of each project (Lobsitz, 1996). In addition, some important challenges in this approach are regarded e.g. how to automate the process of constructing the method (Sellers & Ralyté, 2010) and how to pass the requirements of the organization or project to a semiautomatic mode of identification for optimal collection of elements of the method.

2.3.1.5 Self-adaptive methods

Self-adaptive methods such as XP (Beck & Andres, 2004), Scrum (Schwaber, 1997), Crystal (Cockburn, 2002) and Lean Development define a set of principles, values and practices that in their instantiation allow the configuration a unique process adapted to the needs of the equipment and in general of the project. The process applied to each project emerges the use of these process elements. However, these methodologies subtract the detail of the process that facilitates the reuse of knowledge associated with its execution and depend a lot on people (which is less determinant with the process approach) and the resulting process is not necessarily the most appropriate in the time. In addition, these methodologies emerge from the interpretation and application of the rules defined under the conditions of the project and the people at a particular moment.

2.3.1.6 Families and Process Lines

According to Simidchieva et al. (Simidchieva, Clarke, & Osterweil, 2007) a family of processes is the set of different agreed variations of the same process. Defining a process family can transfer the following benefits to the software process: i) Optimizes efforts in coordination, automation, improvement and training. ii) Allows the reuse in defining new processes. iii) Facilitates the adaptation of large processes through the exchange of components depending on the circumstances of execution. Cass et al. (Cass, Lerner, & McCall, 2000) presented a way to define and represent families of processes with Little-Jil applying the techniques of families of software products to direct the variation of processes.

Rombach (Rombach, 2005) proposes the concept of software process lines as a “way of managing a process and its variants in a systematic way”. This work motived the need for process lines similar to the product lines, so that, processes within an organization can be organized according to similarities and differences, allowing a better adaptation to the needs of a specific project. The vision of SPrL engineering (integrating product lines and process lines) is to group adaptive artifacts and processes in such a way that it can be chosen based on the set of products, process requirements and project constraints.

2.3.1.7 Process Lines using MDE techniques

Model Driven Engineering (MDE) can be used to describe software development methods where its general idea is to create abstract models of software systems and systematically transform them into concrete implementations (France & Rumpe, 2007). MDE can be used with the same idea in process engineering, specifically in adaptation, where it is necessary to model the software processes in general and transform them into processes taking into account the context of the organization and/or project. In addition, MDE provides reuse through a generative strategy, which can be used in software process engineering. In particular, the transformation techniques have been used as strategies of transformation in software process tailoring. Hurtado et al. (Hurtado & Bastarrica, 2009) supports adaptation at the conceptual level and provides mechanisms to make portable and executable adaptation in different process environments. They propose to use MDE techniques to define the organizational processes as models, the adapted process and the context, as well as the process of adaptation as a model of transformation.

2.4 Related Work

2.4.1 SPrL approaches

Washizaki (Washizaki, 2006) proposed a process-tailoring technique which solves the problems with component-based and generator approaches by building a Process- Line Architecture - PLA and deriving project-specific processes from the PLA. This work presents a strategy based on the comparison of process models to define the architecture of a software process line incorporating common elements and variables. The proposal of Armbrust (Armbrust, y otros, 2008) develops a software process line similar to a software product line, it tries to reduce the complexity and the effort required for managing the processes of a software organization. Armbrust (Armbrust, y otros, 2008) defines scoping, modeling and architecting as the main steps in software process line definition and describes in detail the scoping approach based on an analysis of the potential products, the projects expected and the respective process capabilities needed. In addition, the study sketches experience from determining the scope of space process standards for satellite software development.

Ternité (Ternité, 2009) provided an abstract meta-model for process models to be used as a framework for the creation of software process lines. Furthermore, he provides a straightforward example of a specific process line, which is consistent to the presented meta-model. Hurtado et al. (Hurtado J. A., Bastarrica, Ochoa, & Simmonds, 2013) presented an MDE-based approach for automatically tailoring software processes, where transformation rules are used to adapt a general process model to a specific context. They proposed a model-driven approach for software process lines specification and configuration. Moreover, they present two industrial case studies carried out at two small Chilean software development companies. Hurtado et al. (Hurtado & Bastarrica, 2012) presented a concrete SPrL meta-process: Context Adaptable Software Process EngineeRing (CASPER) and its practices. CASPER is a meta-process for building SPrL that includes context modeling, process feature modeling, scope determination, process modeling and production strategy implementation. It supports the adaptation process at a conceptual level and provides mechanisms to executable adaptation in different environments processes. CASPER uses techniques MDE to define as models organizational process, the process adapted and the context, as well as the adaptation process as a transformation model. MDE provides a formal framework for defining the models and transformations required for the automated process tailoring, but like various types of models, it must specify evolution of process families, thus is hard to manage in practice, limiting the industrial adoption of this approach. Trying to increase facility and the industrial applicability for tailoring software process based on MDE strategy, Bastarrica et al. (Bastarrica, Simmonds, & Silvestre, 2014) proposed a megamodel for automated process tailoring. Megamodeling provides an integrating framework for modeling, which enables a controlled evolution of a process family. They provide an industrial case study for illustrating the megamodel definition and describe how this definition eases the coordinated evolution of the whole approach. In previous works (Ruiz & Hurtado, 2013) defined a software process line using CASPER meta-process for providing evidences about the advantages and disadvantages of CASPER approach. We defined a software process line based on Unified Process like a strategy to facilitate its systematic adaptation. Context-Based Process Line-CBPL approach is presented by Magdaleno et al. (Magdaleno, Araujo, Werner, & Alves Batista, 2015); this work presents the software processes composition through process lines. The main goal of this work was to evaluate the feasibility of supporting a process manager in the creation of the organizational process line. This work focuses on phases of analysis and design to create an organizational SPrL, because during these two phases, the process managers need assistance to deal with variability. This approach was developed as an experimental study conducted in the context of a large oil and gas company in Brazil.

Although this study has obtained positive results, it is noteworthy that this is only initial evidence, due to the limited number of participants and the use of only part of the processes of the organization.

Golpayegani et al. (Golpayegani, Azadbakht, & Ramsin, 2013) developed a software process line for Requirements Engineering (RE) in the context of agent-oriented software development; it was a proposed Agent-Oriented Requirements Engineering Process Line (AOREPL), and propose a step-by-step process line engineering approach which enables process engineers to define and instantiate diverse AORE process lines. This work is based on the application of a domain and the application of engineering of Software Product Lines to produce a Software Process Line. The work of Dias De Carvalho et al. (Dias De Carvalho, Chagas, & Reis, 2014) shows the definition of a SPrL considering project planning, project monitoring and control process areas in processes using Scrum agile methodology together with CMMI maturity model. The authors hope that the software process line proposed here be useful to help a software enterprise that uses both of them Scrum and CMMI to create process instances. The limitation of this study was the fact that the SPrL presented had not been empirically evaluated or tested in a real context. Kuhrmann et al. (Kuhrmann, Ternité, Friedrich, Rausch, & Broy, 2016) presented an approach to construct flexible software process lines and show its practical application in the German V-Modell XT standard process. The paper presents an approach to extend a given software process meta-model with software process line capabilities. It extends existing process tailoring instruments by the two concepts: partitioned software process and variability operation. It approaches providing a systematic way to organize process variants within a software process line and to define required modifications of a standard process model.

Teixeira et al. (Teixeira, De Mello, Motta, Werner, & Vasconcelos, 2015) presented a checklist-based inspection technique (PVMCheck) for supporting the detection of defects on SPrL models, especially in process feature models represented using OdysseyProcess-FEX notation. They proposed a checklist-based inspection technique named PVMCheck (Process Variability Modeling Checklist) for supporting the detection of semantic defects in SPrL variability models.

2.4.2 Supporting techniques in SPrL approaches

Variability modeling is a key issue for the adoption of SPrL in industry. There are several factors affecting the adoption of new technologies, such as notation expressiveness and understandability, tool availability, support and usability, and interoperability with standards. Simmonds et al. (Simmonds, Bastarrica, Silvestre, & Quispe, 2013) presented and evaluated the most promising notations for specifying process variability, including both general SPL notations (feature models and OVM) and process domain-specific notations (SPEM and vSPEM). This work analyzes the benefits, drawbacks of two general-purpose (feature models and OVM), and two domain specific (SPEM variability primitives and vSPEM) approaches, as well as discusses what hinders industry adoption in each case. Rouille et al. (Rouille, Combemale, Barais, Touzet, & Jezequel, 2012) proposed an approach to apply the Common Variability Language –CVM, for requirement variability modeling and its binding to the software processes. In this study, the authors perform an experiment (i) to understand how to use this new variability modeling language in this context, (ii) to discuss if CVL enables the management of processes variability, (iii) and to discuss if CVL enables the management of processes variability while being independent of the process metamodel. For its part Oliveira et al (Oliveira, Pazin, Gimenes, & Kulesza, , 2013) presents the SMartySPEM approach, which extends the SPEM profile for representing variabilities in SPrLs taking into consideration the SMarty approach for variability management. SMartySPEM is composed of an UML profile (SMartySPEM-Profile) for representing variabilities and guidelines that contributes to improve the representation and configuration of SPrLs based on SPEM and suggest how to identify variabilities in a SPrL. Aleixo et al (Aleixo, Kulesza, & Oliveira Junior, 2013) carried out a quantitative comparative study to evaluate both the compositional and the annotative approaches for modeling of variabilities from software process lines, in the perspective of the users of these approaches. They defined metrics to evaluate variability management and they conclude that less amount of variability mechanism is better because there is less amount of process elements to manage in the SPrL.

Blum et al. (Blum, Simmonds, & Bastarrica, 2015) proposed the v-algorithm, a process line discovery algorithm. The v-algorithm uses two thresholds to set up a SPrL: highly frequent relation used to build the base process, variable relations define process variability, and rare relations are discarded as noise. The output of this approach is a SPrL represented as a Petri net, which includes information about activity variability. To find a SPrL in this approach, it only takes into account previous projects logs where are not considered potential products neither projects future that will integrate a SPrL. Similarly, Santos et al. (Santos, Oliveira, & Abreu, 2015) proposed a technique to uncover process elements that are candidates to tailoring; the technique is based on the combination of process execution information and mining. They present the variations identification technique, called VarIdentify; the result is the identification of candidate process elements that can be consider like variants of the process. The candidate process elements are discovered by applying process-mining techniques to a project repository that contains execution traces for several process instances. The objective of this work is to pinpoint such candidate process elements, rather than automatically performing tailoring. Meanwhile, the goal of Schramm et al. (Schramm, Dohrmann, & Kuhrmann, 2015) was to understand better software process variability and to analyze the feasibility of the variability operation instrument, and by developing a catalog of variability operations to support process engineers in the systematic and flexible development of process variants within software process lines. However, variability operations are only one instrument among others and, thus, can (and should) be combined with other instruments. The major limitation of this study was to base it on the V- Modell XT only, and the analysis of a specific platform may be hard to transfer to other contexts.

2.4.3 Software process line Scoping

Armbrust et al. (Armbrust, y otros, 2008) describe requirements and concepts for determining the scope of process standards based on a characterization of the potential products, the projects expected for the future, and the respective process capabilities needed. Software process scoping is defined through evaluation and characterization of the process, projects and products. Scoping defines mandatory and optional process parts as determined by the results of the evaluation, and process domain engineering provides the appropriate process model, which reflects the scoping results. In this approach is not clear which decision models can help to determine which process elements should be part of the process line, and which should not. In the same way, Armbrust et al. (Armbrust, y otros, 2009) proposed as major steps in software process line definition to the scoping, modeling, and architecting. Particularly describe the scoping approach that consisting of five main steps: 1) Product analysis, in order to identify product-imposed process needs. 2) Project analysis, in order to identify project-imposed process needs, 3) Process analysis, using the same attributes as for products and projects in order to identify process capabilities. 4) Attribute prioritization, and 5) Scope determination using a mathematical model. The approach identifies redundant and missing processes based on past, present, and anticipated future projects and products of an organization, and assists software process engineers in selecting the right processes for an organization.

Hurtado et al. (Hurtado & Bastarrica, 2012) in their meta-process CASPER, define a process domain engineering: this is an iterative process focused on catch the software process domain knowledge and to develop the process model core assets for enabling the implementation of each context-adapted process model. The process scope in CASPER is defined in an activity that is part of domain engineering called scope analysis where the scope is explicitly defined through a cross reference (or mapping) between process features and context characteristics proposed. The scope is defined through the definition of relations between context attributes and process features; this is a simpler method because the process features are directly related with the context information. SPrL scoping will determine situations where the process will be used and which process elements (common and variables) will be required in each situation. So, planned situations should be related to process features for determining suitable process configurations for each situation (Hurtado & Bastarrica, 2012). In a similar way, in (Ruiz & Hurtado, 2016) the scope process is defined, but it takes into account that in this case the scope is defined in particular context of the a canonical process family based on Unified Process.

2.4.4 Software product lines scoping

Provided that software processes can be considered as software too (Osterweil, 1987), a Software Process Line (SPrL), is a special software product line (SPL) in the software process engineering domain (Hurtado & Bastarrica, 2012). This section presents a systematic literature review and approaches of the SPL scope definition.

2.4.4.1 Systematic literature review

This revision considers the following activities for its execution: planning, conducting and reporting. This SLR considers the following activities for its execution: planning, conducting and reporting. The SPL execution was synthesized, but in *annex I*, it finds all the detailed information of the SPL application.

Planning

Need for a systematic literature review

To characterize and identify according to the literature methods, mechanisms, techniques, and strategies (or related approaches) that are used in scope definition of Software Product Lines in order to verify their characteristics and apply them on scope definition of Software Process Lines. For the above need, a set of research questions was specifically defined, see Table 2. **1**.

Research questions	
Q1. What approaches have been reported in SPL scope definition?	Q1.1 Which methods, mechanisms, techniques, strategies or techniques are used for scope definition?
	Q1.2 How the proposed approaches are validated?
	Q1.3 What is the application context or validation of the approach?
	Q1.4 The approach makes the identification of the range of characteristics that the SPL can cover? (product features)
	Q1.5 Does the approach identify the product parts to be implemented?
Q2. How do the approaches define the scope?	Q2.1 What are kind of scope covered by each approach?
	Q2.2 Is the scope modeled?
	Q2.3 can the scope evolve?
Q3. Do the approaches use some formal nomenclature for scope definition?	Q3.1 What tools and technology of support are used for scoping definition?

Table 2. 1. SRL research questions

Systematic literature review protocol

Data sources

The following data sources were used for SLR development: IEEE Computer Society Digital Library, ACM Digital Library, Science Direct, Scopus and Wiley Online Library. These five data sources were used in order to have a good number of works, taken in account that the greater the number of data sources improvement the possibility of obtaining all existing related works, since there is no unique data source that indexes all existing works at once (Brereton, Kitchenham, Budgen, Turner, & Khalil, 2007).

Search strategy

In the search strategy we identified the keywords with their respective synonyms and plurals. Through combining these keywords and their association using the AND & OR connectors, the search string was developed. Due to the particularities of each of the data sources, it was necessary to define a search specific string for each one. See Table 2. **2.**

Source	Specific search string
IEEE	*(("Abstract":“product line” OR "Abstract":“product lines” OR "Abstract":“product family” OR "Abstract":“product families”) AND (“Abstract:scope OR "Abstract":scoping OR "Abstract":“scoping activity” OR "Abstract":“scoping process” OR "Abstract":“scoping defining” OR "Abstract":“scoping approaches”))*
	The search was executed in the command search and restricted to by abstracts of publications. Publications between 1997-2017
ACM	*recordAbstract:("product line" OR "product lines" OR "product family" OR "product families") AND recordAbstract:("scope" OR "scoping" OR "scoping activity" OR "scoping process" OR "scoping defining" OR "scoping approaches")*
	The search was executed in the advanced search and restricted to by abstracts of publications. Publications between 1997-2017
Science Direct	*(({product line} OR {product lines} OR {product family} OR { product families}) AND ({Scope} OR {scoping}OR { scoping activity} OR { scoping process } OR {scoping defining } OR { scoping approaches }))*
	The search was executed in the advance search and restricted to by abstracts of publications and we select computer Science topic. Publications between 1997-2017
SCOPUS	*ABS (("product line" OR "product lines" OR "product family" OR "product families") AND (“scope” OR "scoping" OR "scoping activity" OR "scoping process" OR "scoping defining" OR "scoping approaches")) AND (LIMIT-TO (SUBJAREA , "COMP "))*
	The search was executed in the advance search and restricted to by abstracts of publications and we select computer Science topic.
Wiley	*("product line" OR "product lines" OR "product family" OR "product families") AND ("scope" OR "scoping" OR "scoping activity" OR "scoping process" OR "scoping defining" OR "scoping approaches") in Abstract between years 1997 and 2017*
	The search was executed in the advance search and restricted to by publications abstracts. Between 1997 and 2017

Table 2. 2. Specific search string

Selection strategy

For making the studies selection, inclusion and exclusion criteria set were defined that allowed to verify their quality and guarantee that they were studies related to the SLR need. The inclusion and exclusion criteria are shown in Table 2. **3.** And Table 2. **4.** respectively.

ID	Inclusion criteria
IC.1	The study addresses SPL scope definition in the context of software engineering, computer science or software development
IC.2	The study discusses some aspect of definition or identification in the SPL scope
IC.3	The study addressing software process lines scoping
IC.4	The study is peer-reviewed in journals, conferences and workshops

Table 2. 3. Inclusion criteria

ID	Exclusion criteria
EC.1	The study addresses SPL scope definition others context different to software engineering, computer science and software development and software
EC.2	The paper addresses exclusively SPL approaches without focusing on Scope aspects
EC.3	The study is not a scientific paper (editorials, prefaces, article summaries, interviews, news, reviews, correspondence, discussions, comments, reader’s letters and summaries of tutorials, workshops, panels, and poster session)

EC.4	The study is written in other languages different than English
EC.5	The study is a systematic literature review
EC.6	The paper was not found

Table 2. 4. Exclusion criterio

SLR execution

Identification and selection of primary studies

The identification and selection of the primary studies were based on two main steps: Step 1 Search in data sources and Step 2 apply inclusion and exclusion criteria.

Step 1: Consisted of applying on each of the data sources the search strings, in this way, the following Table 2. **5**. summarizes the step results.

Digital Library	Results
IEEE	107
ACM	52
Science Direct	17
SCOPUS	180
Wiley	111
Total without debugin process	467
Trash	34
Repeated	78
Total with the debugin process	331

Table 2. 5. First step results

Step 2. In order to reduce the application subjectivity of the inclusion and exclusion criteria, in this step, participated several researchers. In the first iteration of this step, the criteria application was done by reading the following sections of the articles: titles, abstract, key words and conclusions. As a first iteration result, 72 articles were obtained as possible primary studies. In the second iteration, the criteria application was done by reading all the articles content. As the result was obtained a set of 28 articles classified as primary studiesFigure 2. **1**. summarizes the papers review process in this SLR.

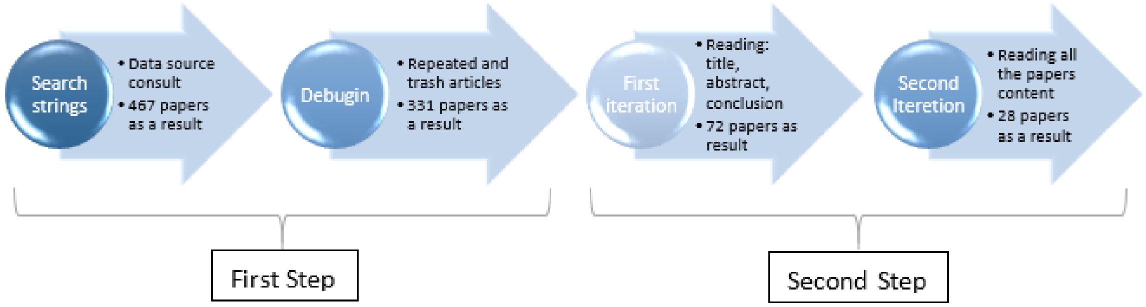


Figure 2. 1. Process papers review

Table 2. **6**. shows the set of 28 primary studies. The data shows that 17 of these studies were published in conferences, 3 in workshops, and 8 in journals.

#	Year	Source	Reference	#	Year	Source	Reference
1	1999	C	(Baud & Schmid, 1999)	15	2011	C	(Balbino, Santana de Almeida, & Meria, 2011)
2	2000	W	(Schmid & Schank, 2000)	16	2011	W	(Cavalcanti, y otros, 2011)
3	2002	C	(Schmid & Verlage, 2002)	17	2011	J	(Heradio-Gil R. , Fernandez-Amoros, Cerrada, & Cerrada, 2011)
4	2002	C	(Kishi, Noda, & Katayama, 2002)	18	2012	C	(Lee & Lee, 2012)
5	2003	C	(Kuloor & Eberlein, 2003)	19	2012	C	(Nöbauer, Seyff, Groher, & Dhungana, 2012)
6	2003	J	(Bosch, 2003)	20	2012	C	(Fernandes, Lemos, & Santana Almeida, 2012)
7	2005	C	(Park & S. D. Kim, 2005)	21	2012	C	(Bartholdt & Becker, 2012)
8	2006	C	(John, Knodel, Lehner, & Muthig, 2006)	22	2012	C	(Gillain, Faulkner, Heymans, Jureta, & Snoeck, 2012)
9	2007	W	(Noor, Grünbacher, & Briggs, 2007)	23	2013	C	(Cruz, y otros, 2013)
10	2009	C	(Wnuk, Regnell, & Karlsson, 2009)	24	2014	J	(Alsawalqah, Kang, & Lee, 2014)
11	2010	J	(Lee, Kang, & Lee, 2010)	25	2014	C	(Nöbauer, Seyff, & Groher, 2014)
12	2010	J	(John I. , 2010)	26	2015	C	(Lanzen, Fontana, Paludo, Malucelli, & Reinehr, 2015)
13	2010	C	(Villela, Dörr, & John, 2010)	27	2016	J	(Alcântara, Brittoa, Andrade, Almeida, & Ayala, 2016)
14	2010	J	(Cvetković & Nešković, 2010)	28	2016	J	(Rossel, Herskovic, & Ormeño, 2016)

Table 2. 6. Primary Studies

Data extraction

In this SLR part, all 28 primary studies were read again, completely, in order to answer our research questions. The data extraction was done through the completion of a spreadsheet which allowed to collect important information of all the primary studies. In this way, studies common attributes as the year, authors, title, source, and event were specified. As part of the extraction strategy, other specifics attributes were defined for answering the research questions. Nine attributes were defined for Q1.1, five for Q1.2, six for Q1.3 and one for Q1.4 and Q1.5. The attributes of Q1.1 refer to a set of classification options for the approaches with respect to their formality of its description. The attributes of Q1.2 is a set of attributes that allow classifying how the approach was validated, and the attributes of Q1.3 indicate the validation context. The attribute of Q1.4 and Q1.5 allow classifying the fulfillment of the scope definition in SPrL. Table 2. **7**. only show the attributes for Q1 but is important to highlight that of the same way was made for all questions.

ID		**Attribute**	**Description**
Q1	Q1.1	Method	Is a method or its parts that are explicitly addressed in the approach described by the primary study?
		Mechanism	Is a mechanism explicitly addressed in the approach described by the primary study?
		Technique	Is a technique explicitly addressed in the approach described by the primary study?
		Strategy	Is a strategy explicitly addressed in the approach described by the primary study?
		Process	Is a process or its parts that are explicitly addressed in the approach described by the primary study?
		Guideline	Is a guideline explicitly addressed in the approach described by the primary study?
		Practice	Is a practice explicitly addressed in the approach described by the primary study?
		Notation	Is a notation explicitly addressed in the approach described by the primary study?
		Other	No mention in done regarding the type approach
	Q1.2	Case study	Is a case study explicitly addressed for the validate of the primary study?
		Small running example	Is a small running example explicitly addressed for the validate of the primary study?
		Several case studies	Are several case studies explicitly addressed for the validate of the primary study?
		Simulation	is simulation explicitly addressed for the validate of the primary study?
		Experiment	is an experiment or quasi-experiment explicitly addressed for the validate of the primary study?
		Validation mentioned but not clarified	is validation mentioned but not clarified in the primary study?
		None	It hasn't validation
	Q1.3	Academic projects	Is an academic project context used to validating of the primary study?
		Real companies' projects	Are a real companies project context used for validating of the primary study?
		Open source projects	Is an open source project context used to validating of the primary study?
		Academic projects and Real companies' projects	Are academic projects and real companies' projects context used to validating of the primary study?
		Other	No mention in done regarding the validate context
		None	It hasn't validation context
	Q.1.4	Characteristics	Are characteristics that the SPL can cover identified in the primary study?
	Q 1.5	product parts	Does the approach identify the product parts to be implemented

Table 2. 7. Classification attributes for Q1

Results and Conclusion

The results show that although the scope in software product lines is a topic that has been addressed since the late '90s, the small number of publications in scientific journals may limit that these approaches provide adequate scientific support to facilitate from this perspective in the definition in SPrL scope. Due to the fact that in the definition of scope process-guided approaches are used, this allows discerning that this same type of approach may be indicated in the SPrL scope definition, but bearing in mind that there are several techniques, practices, guides, and strategies, which could help in the support and definition of more complete technics in such a way that their descriptions are more suitable for the industrial environment providing empirical evidence.

There is a good correspondence in scope definition between SPL and SPrL, this is because the SPrL approach is a particular domain application of a product line and therefore the definition of the scope SPrL can benefit of the different approaches, activities, strategies, and elements used in SPL. In addition, to achieve approaches that allow having a complete reuse infrastructure it is necessary that the scope definition considers the three types of scope (Domain Scoping, Scoping Portfolio, and Asset Scoping), in such a way that they provide a complete baseline for the construction of the reuse infrastructure and generation of processes and process assets in SPrL scoping. Likewise, it is crucial that the scope is materialized in some type of artefact or model in such a way that it adequately allows its analysis, visualization, and improvement. Benefits that will be important in defining the scope in SPrL, allowing a description of the analytically more sophisticated and useful scope for companies. A scope materializes can facilitate its evolution by improving and incorporating elements that amplify or adjust the possibilities of supporting more domain environments and allow support tools use for the execution of this complex activity.

2.4.4.2 SPL scoping Methods

In SPL the activity of determining appropriate bounds to the product line effort is usually referred to as SPL scoping (Schmid K. , 2000). Schmid (Schmid K. , 2000) presented a survey of scoping-related technologies and presented 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 process. In this work the author also defines a categorization of different kinds of scoping, product portfolio scoping, domain scoping and asset scoping. In later work Schmind (Schmid K. , 2002) described an approach to SPL scoping (PuLSE-Eco V2.0) that addresses the different levels of scoping decisions in an integrated manner. This approach supports the levels of domain scoping (as a basis for domain analysis) and reuse infrastructure scoping (as a basis for architecting/ implementation). Schmind's approach fully covers the scoping activities of domain scoping and reuse infrastructure scoping and it was validated in several industrial case studies. For its part, John et al. (John, Knodel, Lehner, & Muthig, 2006) described the PuLSE scoping process of Fraunhofer IESE, which is an update and customization of the PuLSE-Eco scoping process. They define customization factors that influences scoping significantly, which are adaptable for the context. These factors are grouped in five categories: i) operational context, related with project and organizational constraints. ii) Domain characteristics related to domain complexity. iii) Delivery artifacts, this category identifies the existence of artifacts relevant for scoping. iv) Enterprise context, this category is related to the structure and maturity of the organization. v) resources related to the knowledge of the stakeholders and resources available for scoping definition.

Park et. al. (Park & S. D. Kim, 2005) proposes a method for domain analysis and economic analysis for the assets scope, which considers the variability and its dependencies. The proposed approach is based on a systematic process with a set of metrics that help in the definition of the scope effectively in such a way that an economic benefit is produced. Lanzen et. al. (Lanzen, Fontana, Paludo, Malucelli, & Reinehr, 2015) present a semi-automatic approach for the SPLs scope definition. This method is based on the identification and semi-automatic classification of the characteristics of the product, in addition, to provide an approach to evaluate the variability and common points between a baseline and a new product. Other systematic method for SPL scope definition is presented by Kishi et. al. (Kishi, Noda, & Katayama, 2002), in which it is considered that the scope is a decision-making activity in which multiple candidate elements are evaluated and the appropriate ones are selected from point of view of individual optimality and complete optimality. Lee et. al. (Lee & Lee, 2012) propose a method of quantitative reach directed by the market which incorporates in the scope definition the needs of the clients, the structure of the family of products and the market strategies in such a way that guarantees that the derivations of the SPL penetrate in the market networks. For its part Alsawalqah et. al. (Alsawalqah, Kang, & Lee, 2014) proposes a step by step method for the scope optimization that integrates different types of information and techniques. It proposes a mathematical basis and a simulation algorithm to help in the decision-making process in the scope elements selection in an optimized way.

2.4.4.3 SPL scoping Process

In Lee et. al. (Lee, Kang, & Lee, 2010) a comparison and analysis of SPL scope definition approaches is made, in which the common elements are identified and a unified approach is developed with the purpose of being used by the companies that want to plan the launch of a product line. Balbino et. al. (Balbino, Santana de Almeida, & Meria, 2011) show the benefits of combination between SPL and the agile methods. Specifically, an agile scope process for SPL is proposed. For its part Noor et. al. (Noor, Grünbacher, & Briggs, 2007) proposes a collaborative scope approach which is based on involving critical stakeholders to help balance technical and commercial concerns. This approach uses guidelines, collaboration patterns and elements of other successful approaches in scope definition.

PLEvo-Scoping (Product Line Evolution Support at Scoping) (Villela, Dörr, & John, 2010) is a process that helps the teams in SPL scope definition to achieve an anticipation of the emerging characteristics and distinguish the unstable characteristics from the stable ones in order to prepare the product line in the solution of possible adaptation needs. Rossel et. al. (Rossel, Herskovic, & Ormeño, 2016) presents a systematic process to define the domain model and determine the SPL scope in the specific domain of emergency management software. Define formal relationships between artifacts, the roles of each activity and clear conditions of completion for the scope definition. Other work that present an approach in specific domain is presented by Cvetković et. al. (Cvetković & Nešković, 2010) which aims to identify the scope of families of software products in the telecommunications domain. In this approach, a generic architecture of the families of products is presented, as well as a methodological procedure for the identification and definition of Scope. In Bartholdt ed. al. (Bartholdt & Becker, 2012) describes the best practices to make extension of the scope sustainable in the long term in which various types of management means are used. In particular, it describes a way to identify subdomains and maturity concerns to consider when deciding on reuse; In addition, a strategy for dividing existing assets into common modules and specific extensions of the product line is described. CAVE (Commonality and Variability Extract) approach (John I. , 2010) is a solution that try to face to the effort required in scope definition, in this several experts participate per subdomain of the SPL, which can implies investing a lot of time in workshops and interviews for the information collection that allows make an adequate SPL definition. This approach is based on information systematic collection from the existing systems documentation for scope definition, which can reduce the time spent by experts, allowing the introduction of the SPL to be much faster.

Alcântara ed.al. (Alcântara, Brittoa, Andrade, Almeida, & Ayala, 2016) present an improved version of a hybrid approach proposed to solve the selection problem of the characteristics model, in order to support the definition of the product portfolio scope. It is an independent approach of algorithms and technology using two tools that improve the collection of the required information. In the same way. Cruz et. al. (Cruz, y otros, 2013) define a approach focuses on the portfolio scope definition, which takes into account the different needs of clients to obtain an optimized products portfolio. This approach makes an important correlation of the characteristics perceived by the clients in the source code in a systematic way and includes an optimization module to generate the SPL products candidate.

2.4.4.4 SPL scoping Techniques

The study presented by Baud et. al. (Baud & Schmid, 1999) is focuses on the portfolio scope definition for SPLs which considers the different needs of customers to obtain an optimized products portfolio. This approach makes an important correlation of the characteristics perceived by the clients in the source code in a systematic way and includes an optimization module to generate a scope candidate product. Wnuk et. al. (Wnuk, Regnell, & Karlsson, 2009) present the Feature Survival Charts for visualization of scoping change dynamics technique for visualizing the dynamics of change in the scope of three projects in a large-scale industrial environment. The technique allows to effectively investigate the reasons behind the scope decisions which can be valuable for future improvements.

An approach supported by technological tools for the semi-automatic analysis of an existing products family is proposed by (Nöbauer, Seyff, & Groher, 2014). It uses the calculation of their similarity, allowing the initial estimation of the reuse potential with little effort and thus supporting traditional or manual activities of the scope definition. Gillain ed. al. (Gillain, Faulkner, Heymans, Jureta, & Snoeck, 2012) proposes a mathematical technique capable of optimizing the SPL scope and sketch both the development and the launch planning. Gillain too states that the identification and evaluation of the scope types (product portfolio, domain and assets) can be done separately, but their optimization must be done in an integrated way.

2.4.4.5 SPL scoping Others approaches

Schmid et. al. (Schmid & Schank, 2000) described PuLSE-BEAT as a tool to support the realization of the scope of the PuLSE-Eco approach. It describes the main needs that a support tool for the definition of the scope must support and also shows how PuLSE-BEAT supports each of these needs. John et. al. (John, Knodel, Lehner, & Muthig, 2006) presents an update and customization of the PuLSE-Eco scope process. Explicitly integrates 21 personalization factors that significantly influence scope definition allowing more appropriate scope definitions.

Bosch (Bosch, 2003) presents the scope notion, its creation phases, maturation and collapse for what organizations normally face is discussed. The hierarchical nature of the scope is discussed, that is, a scope is defined within a broader scope. Especially in larger organizations, there may be several scope levels. To illustrate multiple-scope approaches, product lines and approaches to product populations are discussed. On the other hand, Heradio-Gil ed. al. (Heradio-Gil R. , Fernandez-Amoros, Cerrada, & Cerrada, 2011) discussed the SPL scope importance and proposed algorithm to calculate common points between the characteristics diagrams in a quadratic time. Specifies an algorithm for abstract notation for feature diagrams, called NFT, for which an abstract syntax and its NFT semantics have also been formally defined.

In Nöbauer et. al. (Nöbauer, Seyff, Groher, & Dhungana, 2012) shows a light scope approach for SMEs which helps to obtain a quick feedback on the reuse potential of existing products, it is a conceptual solution that allows companies to identify semi automatically the similarity within existing product configurations. Fernandes et. al. (Fernandes, Lemos, & Santana Almeida, 2012) proposes a tool to support the scoping process based on the existing products source code, in order to reduce costs and time involved in the process. In another way, Cavalcanti et. al. (Cavalcanti, y otros, 2011) presents a meta-model whose main objective is to support and coordinate the SPL phases, defines five sub-models: project and risk management, scope, requirements and tests. It proposes to make an adequate management of these phases and to define responsible, and to maintain the traceability and variability between the different artifacts to make an adequate SPL management.

Kuloor et. al. (Kuloor & Eberlein, 2003), proposed a systematic and iterative Requirements Engineering approach for product lines development with a specific technique for each stage. This approach includes the activities necessary for correct identification, analysis, modeling and specification of the SPL requirements. It proposes the use of several techniques such as aspect-oriented programming, which is used to analyze common and variable requirements, product maps, which are used to determine the scope and product family characteristics, and XML is used, as a document template for specifying and navigating various product line artifacts.

Clements (Clements P. , 2002) explored the issues associated with scope, and its importance in the overall success of the product line. Moreover, in this work, he also makes some observations about the effects that a well-defined scope has on the product line organization and the way it handles related issues such as the architecture for the product line, its customer interface, and how it reacts to a new product opportunity. The author also makes a distinction between reactive scoping (the norm with most organizations) and pro-active scoping (which an organization can employ to achieve great strategic advantage). Moraes (Moraes, Santana de Almeida, & Romero de Lemos Meira, 2009) presented a systematic review to investigate the existent approaches on software product line scoping, aiming at identifying, comparing and summarizing evidence about the scope definition techniques, analyzing their activities, roles, guidelines, concepts, strong points and drawbacks, and the main features. In addition, the review presents an important result 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 relate works in scoping approaches, showing how scoping is addressed by the approaches.

The following tables summarize the characterization of previous studies both in the context of process lines and products. **Table 2. 8**. shows the comparison of the approaches in process lines, for compliance with the phases defined in the Software Process Line Engineering. The comparison is made by fulfilling each of the phases using for this the assignment of the following values: “+”, which indicates that the approach considers the phase explicitly and its description is clear; "-" indicates that the approach does not consider the phase, "or" indicates that the approach considers the phase but its description is not clear. **Table 2. 9**. summarizes the characterization of the studies in SPLs using attributes that allowed the extraction of information from each of the approaches. This table was obtained by executing a systematic review of the literature, see Annex I.

	Construction			Usage		
Authors	Scope	Modeling	Architecture	Instantiation	Customization	Project Specific Process
(Washizaki, 2006)	-	+	+	o	-	-
(Armbrust, y otros, 2008)	+	-	-	-	-	-
(Ternité, 2009)	-	-	+	o	-	o
(Hurtado J. A., Bastarrica, Ochoa, & Simmonds, 2013)	-	o	-	+	O	+
(Hurtado & Bastarrica, 2012)	O	O	O	+	O	+
(Bastarrica, Simmonds, & Silvestre, 2014)	-	+	-	+	-	+
(Ruiz & Hurtado, 2013)	O	O	O	+	-	+
(Magdaleno, Araujo, Werner, & Alves Batista, 2015)	O	+	+	-	-	-
(Golpayegani, Azadbakht, & Ramsin, 2013)	o	+	o	+	o	+
(Dias De Carvalho, Chagas, & Reis, 2014)	-	+	+	-	-	-
(Kuhrmann, Ternité, Friedrich, Rausch, & Broy, 2016)	-	+	+	o	-	o
(Teixeira, De Mello, Motta, Werner, & Vasconcelos, 2015)	-	+	-	-	-	-
(Simmonds, Bastarrica, Silvestre, & Quispe, 2013)	-	+	-	-	-	-
(Rouille, Combemale, Barais, Touzet, & Jezequel, 2012)	-	+	-	-	-	-
(Oliveira, Pazin, Gimenes, & Kulesza, , 2013)	-	+	-	-	-	-
(Aleixo, Kulesza, & Oliveira Junior, 2013)	-	+	-	-	-	-
(Blum, Simmonds, & Bastarrica, 2015)	-	+	-	-	-	-
(Santos, Oliveira, & Abreu, 2015)	-	+	-	O	-	-
(Schramm, Dohrmann, & Kuhrmann, 2015)	-	+	-	-	-	-
(Armbrust, y otros, 2009)	+	-	-	O	O	O

Table 2. 8. Characterization of the process lines approaches

Authors	Approach	Validated type	Scope level	Modeled	Scope evolve	support tools	Application context or validation	Tasks or activities
(Schmid K. , 2002)	Method	Simulation	DS & AS	NO	NO	No tool mentioned	Real companies projects	YES
(Park & S. D. Kim, 2005)	Method	Case study	DS & AS	NO	NO	No tool mentioned	Academic projects	YES
(Balbino, Santana de Almeida, & Meria, 2011)	Process	None	PPS, DS, AS	NO	NO	No tool mentioned	None	YES
(Nöbauer, Seyff, Groher, & Dhungana, A Lightweight Approach for Product Line Scoping, 2012)	Other	None	PPS & DS	NO	NO	Prototype tool	None	YES
(John, Knodel, Lehner, & Muthig, 2006)	Guideline	Not clarified	PPS,DS, AS	NO	NO	No tool mentioned	None	YES
(Kishi, Noda, & Katayama, 2002)	Method	Case study	PPS, DS, AS	NO	NO	No mentioned	Real companies projects	YES
(Alsawalqah, Kang, & Lee, 2014)	Method	Case study	PPS	YES	YES	No mentioned	Academic projects	YES
(Lee & Lee, 2012)	Method	Case study	PPS	NO	NO	No mentioned	Open source projects	NO
(Alcântara, Brittoa, Andrade, Almeida, & Ayala, 2016)	Process	Case study	PPS	NO	NO	Prototype	Open source projects	NO
(Baud & Schmid, 1999)	Technique	Small running example	PPS	NO	NO	No mentioned	Real companies projects	YES
(Lee, Kang, & Lee, 2010)	Process	Case study	PPS,DS, AS	NO	NO	No mentioned	Other	YES
(Fernandes, Lemos, & Santana Almeida, 2012)	Practice	None	AS	NO	NO	Prototype	None	NO
(Bartholdt & Becker, 2012)	Process	None	DS & AS	NO	YES	No mentioned	None	NO
(Schmid & Schank, 2000)	Strategy	Small running example	PPS,DS,AS	NO	NO	Prototype		
(Lanzen, Fontana, Paludo, Malucelli, & Reinehr, 2015)	Method	Simulation	DS	NO	NO	Proposal	Real companies projects	YES
(Noor, Grünbacher, & Briggs, 2007)	Process	None	DS & AS	NO	NO	Proposal	Open source projects	YES
(John I. , 2010)	Process	Several case studies	DS & AS	NO	NO	No mentioned	Real companies projects	YES
(Wnuk, Regnell, & Karlsson, 2009)	Technique	Case study	DS & AS		NO		Real companies projects	YES
(Cavalcanti, y otros, 2011)	Notation	Not clarified	AS	YES	NO	Available online	Real companies projects	NO
(Villela, Dörr, & John, 2010)	Process	Quasi-experiment	PPS,DS, AS	NO	YES	No clarification	Academic projects	YES
(Cvetković & Nešković, 2010)	Process				NO	No mentioned		YES
(Kuloor & Eberlein, 2003)	Practice	Small running example	PPS, AS	NO	NO	No mentioned	None	NO
(Gillain, Faulkner, Heymans, Jureta, & Snoeck, 2012)	Technique	Case study	PPS, AS	YES	no	No mentioned	Real companies projects	NO
(Nöbauer, Seyff, & Groher, 2014)	Technique	Case study	DS & AS	NO	NO	Prototype	Real companies projects	NO
(Heradio-Gil R. , Fernandez-Amoros, Cerrada, & Cerrada, 2011)	Other	Other	AS	YES	NO	Supported by community	Other	NO
(Rossel, Herskovic, & Ormeño, 2016)	Process	Not clarified	PPS & DS	YES	no	No mentioned	Real companies projects	YES
(Cruz, y otros, 2013)	Process	Case study	PPS	NO	NO	No clarification	Real companies projects	YES
(Bosch, 2003)	Strategy	None	AS	NO	YES	No mentioned	None	NO

Table 2. 9. SPL scoping characterize.

2.4.5 Synthesis and discusión

Each section in this chapter provides a broad information about concepts, approaches, techniques, and strategies related to the software process tailoring, process lines, and product lines. These works presented the needs for adapting the development processes through of several approaches where each one resolves the tailoring problem using different perspectives. Software process line, is a special software product line in the software process engineering domain (Hurtado & Bastarrica, 2012) and that it is one of the best strategies in software process tailoring, because that processes can be organized according to similarities and differences in a planned way, facilitating their management, reuse and adaptation to the project specific needs. The work comparison as it is described above in table 1, show the need for defining approaches for tailoring software process in SPrL context, but none of the works presented here specifically defines an adequate approach as does SCOPE (Armbrust, y otros, 2009) and partially CASPER (Hurtado & Bastarrica, 2012) for scope definition in process lines; this is a starting point for intending to unify concepts and for establishing a specific path to build and apply the scope definition. Moreover, the Table 2, characterizes and identify according to the literature methods, mechanisms, techniques, and strategies (or related approaches) that are used in scope definition of Software Product Lines in order to verify their characteristics and apply them on scope definition of Software Process Lines. In this work, we propose a hybrid scope approach, which is based on the identification of the specific needs expressed by the products and projects with respect to the processes that make up the line, providing a clear and guided approach for its realization. The identification of the needs of context, with respect to the elements of the process, is done through an indicator of suitability that allow to establish a correlation between these two elements (context and process) in a way that allows to identify and select the process elements that will be part of scope line facilitating by the tailoring processes to the specific needs. Furthermore, this work prove evidence empirical of SPrL application specifically over the scope determination in industrial contexts where this activity is a key and complex activity that influence in the successful of SPrL approach.

Chapter 3

Scope determination in Software Process Lines – SpeTion - SPrL

3.1 Overview

This chapter shows the solution idea behind SpeTion-SPrL. Likewise, It is presented to SpeTion-SPrL how a systematic approach for scoping determination where its tasks are detailed.

3.2 SpeTion-SPrL Solution idea

The solution idea includes several elements defined both for the context of product lines and process lines. Among these elements there are some common and others that are specific to each context. These elements are shown in **Figure 3.1**.

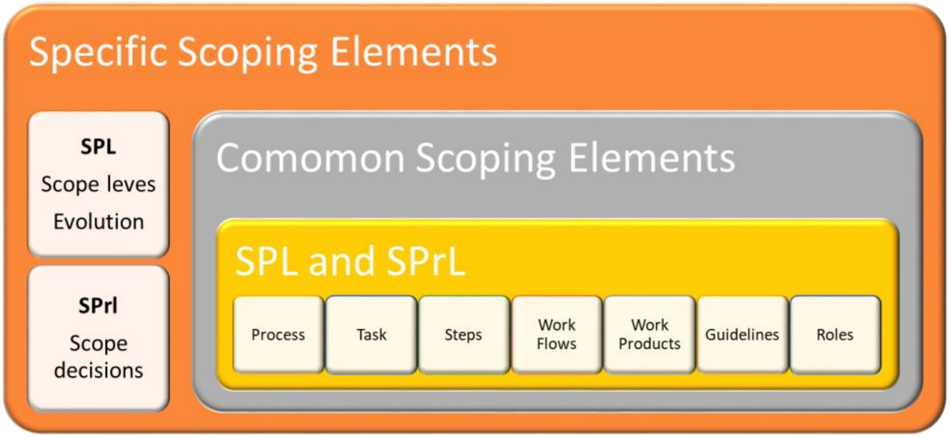


Figure 3.2. SpeTion –SPrL Solution Idea elements

Common Scoping Elements

a) *Process:* SpeTion-SPrL contains two processes to define SPrL´s scoping in a systematic way, it enables to homogenize the scope building, supports the materialization and standardization of the scope elements and it can improve the knowledge of those involved in the process.

b) *Task and steps:* SpeTion-SPrL contains a set of the process elements which are the basic elements for building the SpeTion-SPrL process. It describes a basic work unit.
c) *Workflows:* SpeTion-SPrL contains a set of workflows which are the operational aspects of an activity*.*
d) *Work products:* It are process elements that are tangible work products consumed, produced, or modified by tasks. They may serve as a basis for defining reusable assets in the scoping process.
e) *Guidelines*: It is a process element that directions in detail the execution of some aspect of the SpeTion-SPrL approach
f) *Roles:* It is a process element that defines a set of related skills, competencies, and responsibilities.

Specific Scoping Elements

a) *Scope levels*: SpeTion-SPrL consider as a part of the scope definition the use of the product scope and assets scope according to SPL context. These elements are materialized with the identification the projects and products, and process assets respectively in the process context
b) *Evolution:* SpeTion-SPrL allow the scope evolution through on a specific task and tangible scope matrix through which the scope can be extended or adapted
c) *Scope decisions:* SpeTion-SPrL allows decision making through the scope matrix and its suitability value. This allows *adaptation rules definition*, where scope decisions are materialized, to include or remove or add of process assets.

3.3 ScoPE determinaTION in Software Process Lines – SpeTion- SPrL

3.3.1 Introduction to SpeTion- SPrL

In recent years, different approaches have emerged as an alternative for adapting the software process. One of the strategies in tailoring software process is the Software Process Lines - SPrL. A Software Process Line is a set of processes in a particular domain or for a particular purpose, which has common characteristics and is based on common reusable process assets" Washizaki et al. (Washizaki, 2006). SPrL is an approach that allows the tailoring and evolution of software processes. To define the processes and the assets of the processes to be used and the situations that demand these processes, normally in the SPrL definition an activity will be carried out to determine the scope. In the SPrL scope definition is necessary to identify the range of characteristics that the SPrL processes must cover, as well as determine the situations in which the process will be used and the process elements (common and variable) that will be required in each situation Armbrust et al. (Armbrust, y otros, 2009). Scope determination is a key activity that facilitates or hinders organizations from achieving an effective SPrL solution.

This thesis proposes SpeTion – SPrL (Scope DeterminaTion in Software Process Lines), an approach for SPrL scoping. SpeTion-SPrL facilitates scope definition in a systematic way, this approach uses coherently different aspects both for the context of product lines and process lines. SpeTion-SPrL was built following the methodological support of Situational Method Engineering (SME) in an adapted way (Ralyté, Deneckère, & Roll, 2003), which involved the execution of two iterations in the SME steps, Specification of the method requirements ,selection of the method components, and assembly of the selected method components, See annex H.

3.3.2 SpeTion – SPrL principles

SpeTion-SPrl is an approach for scoping determination, which is based on the identification of the specific needs expressed by the products and projects with respect to the process assets. The identification of the needs will allow a correlation with the process assets in a way that allows identifying and selecting the variable aspects for deriving adapted processes. SpeTion-SPrL has been built on the following four principles, see Figure 3. **3.**, and which are supported by SpeTion-SPrL, how Table 3. **1.** show it.

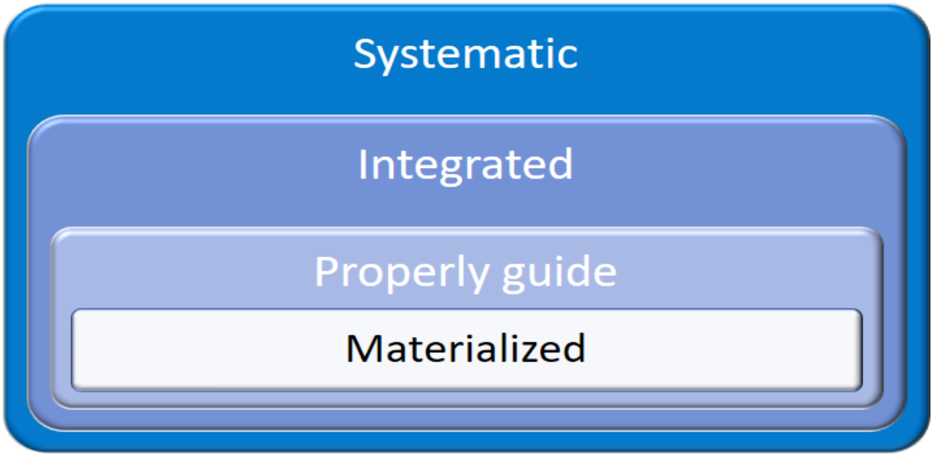


Figure 3. 3. SpeTion – SPrL principles

a) *Principle 1. Systematic:* the approaches in scope definition sometimes forget that a scoping definition method must be systematic (Schmid K. , 2000), (Armbrust, y otros, 2009), (Carvalho & Chagas, 2014). SpeTion-SPrL provides elements such as: process, tasks, work products, guidelines and roles, identifies scope levels and suitability levels so that the results of their execution are systematic.

b) *Principle 2. Integrated:* The SPL have proven successful in the software industry, therefore, it is necessary to take them into account to incorporate their best practical in the SPrL context. SpeTion-SPrL uses elements such as: Scope levels, scope evolution, scope decisions, tailoring rules, work products like integrated elements between SPL and SPrL approach. Moreover, to integrates some parts of SCOPE and CASPER approach

c) *Principle 3. Properly guide:* SPrLs scope definition approaches does not have a clear direction so that the software industry can make its incorporation in an appropriate way (Armbrust, y otros, 2009), (Hurtado & Bastarrica, 2012). SpeTion-SPrL provides elements such as: scope levels, process, tasks, work products, guidelines and roles so that the scope definition is properly driven

d) *Principle 4. Materializable:* The scope does not have clear guidelines and structured documents supporting its specification, which is why it can be considered ambiguous and difficult to communicate (Ruiz, Camacho, & Hurtado, 2018). SpeTion-SPrL provides work products and applied scope levels for understanding scope concept and make it tangible and materializable.

	Scope Levels	Evolution	Scope decisions	Tailoring Rules	Process	Activities, task, steps	Work products	Guidelines	Roles
Systematic	X				X	X	X	X	X
Integrated	X	X	X	X			X		
Properly driven	X				X	X	X	X	X
Materialize		X					X		

Table 3. 1. SpeTion-SPrL Principles support

SpeTion – SPrL approach uses different aspects both for the context of product lines and process lines, but it incorporates its own techniques such as techniques interviews, surveys, focus groups and workshops for extracting the scope knowledge, and a scope quantified cross matrix.

3.3.3 Scope sceneries

An approach for SPrL scope determination must consider two scenarios, one for its creation and the other for its use. The first scenario occurs in the SPrL construction, specifically in the scope definition, where the scope is analyzed and define the scope by identifying the process suitability (supply) and process features face to typical situations of the process line. For supporting this scenario, SpeTion-SPrL proposes several tasks guiding the analysis and design of the scope. Similarly, it identifies the possible situations that the process and their elements must adequately support.

The second scenario is the scope use, which takes place when it is necessary to determine if the process line can support a new situation (demand), in addition, to determine if the available process assets that can meet the specific needs of this new situation. To support this scenario, several tasks are described for guiding the scope document use. The activities are illustrated in the next sessions.

3.3.3.1 Process of analysis and design of the Scope

This process supports the scope creation scenario, view Figure 3.**4**., for this, it defines the following tasks: *Identify projects, identify products, identify situations, to elicit of software process features* and *suitability determination*. *Identify projects* aim to find software projects that the company has under execution or plans to execute. The *identify product* has the objective of recognizing the current software products that the company is developing and the future ones it intends to develop. The search for projects and products must be developed for a specific domain; As an identification result, an overview of the demands that the process must support will be obtained. The *identify situations* is given by the characterization of the projects and products that the SPrL must support. A situation is defined as the characteristics set that expresses the needs of the projects or products with respect to the process. *To elicit* of *the process features* has the aim to find the process elements, both common and variants. In the *suitability determination*, a suitability measures set of the process features must be established, so that, through quantitative criteria, the inclusion or exclusion of the scope elements can be determined in precisely and planned way. For each process feature, its level of suitability must be defined and then used as a selection criterion according to each situation.

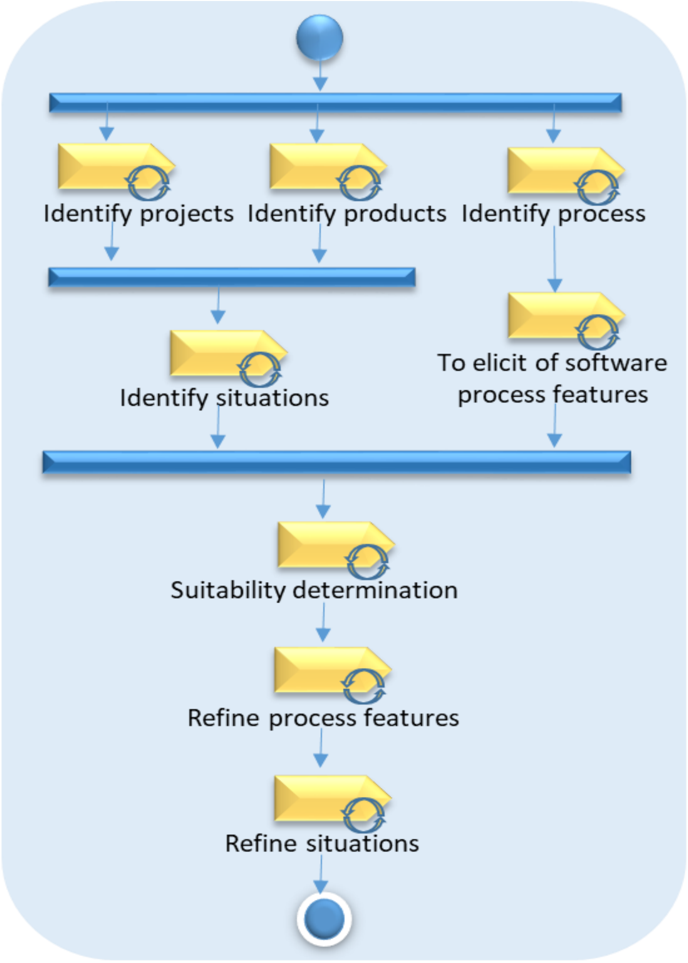


Figure 3.4. Process of analysis and design of the Scope

Identify projects

To know and identify the projects that the company is developing (current) or intends to develop (futures), it is necessary to look for information on the documentation that the company has available and, it too is possible to consult the enterprise staff who can provide information about the projects. This task is support by techniques interviews, surveys, focus group.

Objective: Identify the software projects that the company is developing or intends to develop

Some of the information sources that can be consulted are the following:

- ✓ Project documentation
- ✓ Products
- ✓ Company vision
- ✓ Product portfolio
- ✓ Project management platforms
- ✓ All documentation generated in the development of projects or products

Steps:

1) *Identify a specific domain*
2) *Collect information or inquire about project information*
3) *Select the set of projects according to the domain*
4) *List the set of projects*

Output: Document with the projects list

Identify products

In order to know and identify the software products that the company is developing (current) or intends to develop (future), it is necessary to look for information on the documentation that the company has available and, it too is possible to consult some people who can provide information about the products. This task is support by techniques interviews, surveys, focus group.

Objective: To identify, in a specific domain, the software products that the company is developing or intends to develop

Some of the information sources that can be consulted are the following:

- ✓ Software products
- ✓ Product information (requirements documentation, design, tests, manuals)
- ✓ Project management platforms
- ✓ Company vision
- ✓ Product portfolio. etc.

Steps:

1) *Identify a domain (must be the same project domain)*
2) *Collect information or inquire about product information*
3) *Select the products set according to the domain*
4) *List the products set*

Output: Document with the products list

Identify situations

A situation is defined as a set of context characteristics that expresses the needs of the projects or products with respect to the process. The situations identification is given by the projects and products characterization that the process line must support. For each of the product or project characteristic identified it is also necessary to establish a possible values set. This task is support by Context Definition Method

Objective: Identify situations through the characterization of projects and products

Some of the information sources that can be consulted are the following:

- ✓ List of projects and products
- ✓ Information on the project and product restrictions
- ✓ Company documentation that allows obtaining information about the characteristics of the projects and products

Steps:

1) *Identify the set of characteristics of the projects that require that the process must support them.* The characteristics that demand special attention by the process must be identified. Characteristics list for the analysis of products and projects will depend on the product and organization type. The characteristics list of the projects must be common to the set of characteristics of the projects

   Project characteristics example

   i. Project execution time (PT), refers to the time period the company has to develop the project.
   ii. Project experience (PE), refers to the experience that the company has in the development of a project type

2) *For each set of characteristic, the meaning of the possible values that the characteristic can take.*

   Example of the meaning of three characteristic values for the previous characteristics.

Characteristics	Value 1	Value 2	Value 3
PT	High Execution Time (HET)	Medium Execution Time- MET	Low Execution Time (LET)
PE	High Project Experience (HPE)	Medium Project Experience (MPE)	Low Project Experience (MPE)

Table 3. 2. Meaning example of the characteristic values

3) *Identify the set of characteristics of the products that require that the process must support them,* that is, the characteristics that demand special attention from the process. The features list for product analysis will depend on the product and organization type. This step and the next one is done in a similar way to step 1 and 2 respectively, but considering that this step is focused on the products

Output: Characteristic list

Identify Process

The company can have several processes to support the different projects that it develops. This activity identifies and selects the software process that the company has available for a specific context.

Objective: To identify software processes the company has available and select one for a specific context

Inputs:

- ✓ Process description documents
- ✓ Company documentation that allows obtaining information about the processes

Steps:

1) Identify available processes

2)Collect information or inquire about processes information

3)Select the process according to the project domain

Output: Specific process

To elicit of software process features

This activity defines and identifies process variability which will provide flexibility, that is, the common and variables elements of the process are identified. Figure 3.**5**., shows an example of process variability. Where the process consists of 5 tasks, of these 4 tasks are mandatory and one is optional. The figure contains three variation points, the first is task 2, which can take either of the two alternatives for the task development. The other variation point is task 3, which can take any of its four alternatives. The final variation point is given by the option of task 4, the option refers to a task may or may not be part of the process. This task is support by process comparison techniques like is described in (Washizaki, 2006) or focuses group for knowing the features process

Objective: Define and identify the process features

Inputs:

- ✓ Process description documents
- ✓ Processes list

Steps:

1) *Identify the process common elements.* This step can be done by comparing the elements of the processes that are part of the line. The elements repeated in the comparison can be considered as common and mandatory elements
2) *Identify variation points.* From the comparison made in the previous step is possible to identify variation points, that is, identify process parts where it can have different routes of execution.
3) *Identify the process variants.* The variants of the process are those possible alternatives where the process has a point of variation.
4) Define a process variability model.

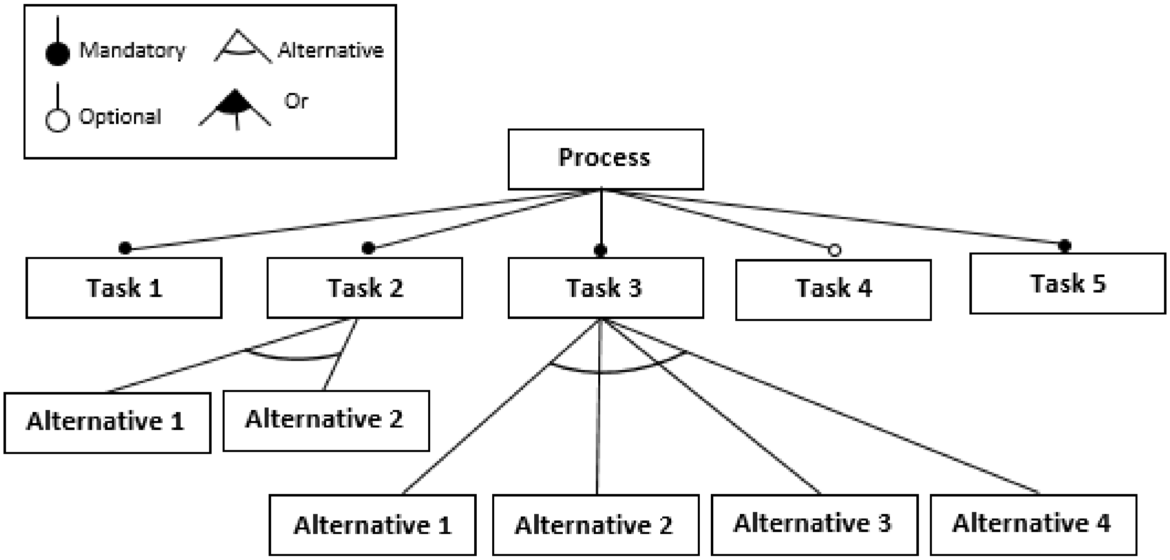


Figure 3.5. Process variability model

Output: Process variability model

Suitability determination

A set of suitability measures of the process variant elements must be determined, so that, by quantitative criteria, the inclusion or exclusion of the scope elements can be accurately planned and determined. For each process variant, the suitability level must be calculated and then used as a selection criterion according to each situation. The suitability level is represented, by the ability of the variable elements of the process to support a certain value of the characteristics specified in a situation. In other words, Table 3. **3**. the task TX has a suitability level of "Inadequate" when the "Characteristic A" takes the value1, which determines that this task does not support this characteristic with this value.

Suitability	**Value**
Totally inappropriate (TI)	0
Inadequate (I)	0.25
Neutral (N)	0.5
Suitable (S)	0.75
Totally adequate (TA)	1

Table 3. 3. Suitability values table

Characteristics	**Values**	**Variant tasks by variation point or by optionality**		
		TX:	**TY:**	**TZ:**
Characteristics A	Value1	I	TA	S
	Value2	I	S	N
	Value3	TA	TA	S
Characteristics B	Value1	N	S	S
	Value2	N	S	N
	Value3	I	TA	TA
Characteristics C	Value1	I	N	S
	Value2	N	S	N
	Value3	I	TA	TI

Table 3.4. Scope Matrix

Columns 1 and 2 of Table 3.**4.** correspond to the characteristics and values identified in the *Identify situations task*. Columns 3, 4 and 5 refer to the process variants of each of the variation points. Each value is determined by values between 0 and 1 corresponding to {TI, I, N, S, TA}, view table 3.3, the values close to 0 (TI) defines that the variable element of the process is not suitable to be part of the process and the values close to (TA) defines that the element has a good level of suitability, therefore, it can be part of the process.

Objective: To determine a measure of the suitability of the process variant elements

Inputs:

- ✓ Process characteristics
- ✓ Features List

Steps:

1) *Define the level of suitability:* Taking into account the company processes information makes the following steps for each process variable element
    a) Define the suitability level depending on the values that a characteristic can take. The suitability level is established between the value of the context characteristic (rows) and the corresponding variable process feature (column). See example in Table 3.4
    b) It may be that there is no precise information for the suitability determination, for these cases, a consensus can be reached and determined by experience its value. In cases where a consensus is not reached or information is not available, it must be established as indeterminate suitability, that is, a hyphen (-) can be placed.

Output: Scope matrix with suitability indexes

Refine Process Features

This task studies the suitability values that each variation point takes in a specific context in order to establish if the variation points are maintained or may become a common element or not be part of the process line.

Objective: To refine variation points in scope matrix

Inputs:

- ✓ *Scope matrix*

Steps:

1) Search among the columns of the scope matrix that have mostly equal or similar suitability values. For these columns, consider the following:

   a) *If the suitability values of these columns are "TA",* it may be that this process element is no variation point to become a common element in the process. In this case, the process element would come out of the SPrL variability and in this way the scope is reduced.

   b) *If the suitability values of these columns are "TI",* it may be that this element of the process is no part of the process line, because it is totally inappropriate for any situation. In this case, the process element would leave the SPrL and in this way the scope is reduced.

Output: *Scope matrix*

Refine Situations

This task studies the suitability values that each characteristics takes in a specific situation in order to establish if these values are not affecting the variation points to remove this value from the characteristics and thus reduce the possible situations

Objective: To refine situations of scope matrix

Inputs:

- ✓ *Scope matrix*

Steps:

1) Search among the rows of the scope matrix those that have mostly equal or similar suitability values. For these rows, consider the following:
   a) if the values are N, it may be that this characteristic value is not affecting the variation points, in this case, it can be considered to remove this value from the characteristics and thus reduce the possible situations. In this case, the SPrL scope is being reduced.
   b) If the values are TI, it may be that this characteristic value is not affecting the variation points, in this case, it can be considered to remove this value from the characteristics and thus reduce the

possible situations. In this case, the scope of the SPrL is being reduced.

Output: *Scope matrix*

3.3.3.2 Process of Scope use

This process supports the scope use scenario, view Figure 3. **6**, the results of *Process of Analysis and design of the Scope* are used, that is, it seeks to verify whether the SPrL supports some new situations; and also help identify the possible assets from a particular process to address a new situation. This process defines the following tasks: *Specify the situation, Process scope evaluation, Customization and Evaluation*. *Specify the situation task* has the objective of to establish the set of characteristics that allow defining a new situation. The propose of *Process scope evaluation task* is to verify if there are enough process assets to support a new situation. *Customization* aims to study the realization of adjustments within the scope in case there are not enough process assets to face a specific situation. *Evaluation task* has the purpose to determine the reasons why the process line can or cannot support a new situation.

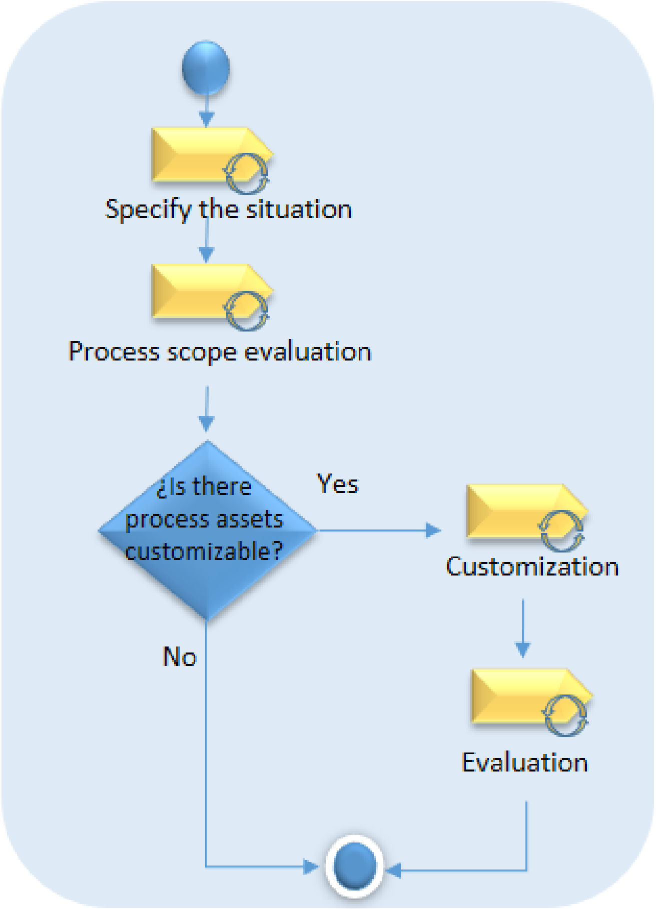


Figure 3. 6. Process of scope use

Specify the situation

Once the set of characteristics has been defined, in a general way, in a specific domain, specific situations can be defined to verify if the SPrL can support a concrete situation.

Objective: To establish characteristics set that allows defining a specific situation

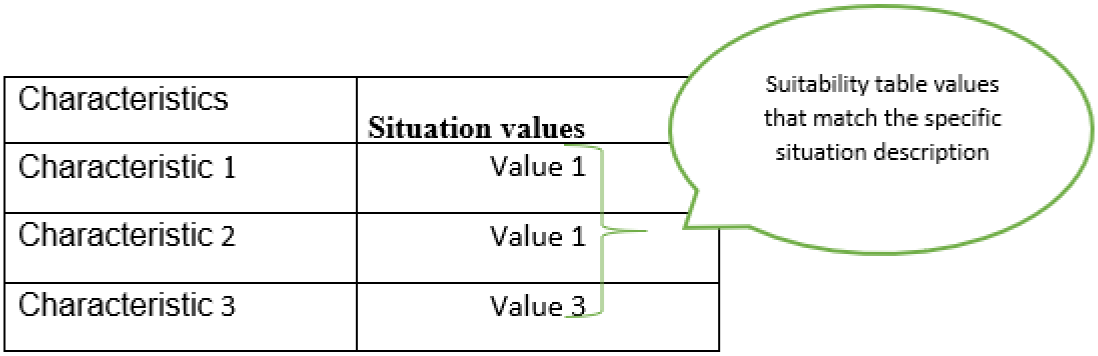
Inputs:

- ✓ List of features
- ✓ Specific situation

Steps:

1) Having a situation, a table with the list of these characteristics and their values that describe the situation must be filled out. The situation characteristics must be the same as the characteristics identified in the identify situations task. Table 3. **5**. is an example of the situation characteristics list, where the first column is the characteristics list and the second column correspond to the suitability values, Table 3.**4**. Scope Matrix, that representing the specific situation characteristics.

Example



Characteristics	**Situation values**
Characteristic 1	Value 1
Characteristic 2	Value 1
Characteristic 3	Value 3

Suitability table values that match the specific situation description

Table 3. 5. Situation characteristics list

Output: List of specific characteristics

Process scope evaluation

The scope evaluation is done by comparing and evaluating the final suitability values obtained by each process variant in a specific situation

Objective: To evaluate the process assets suitability for supporting a new situation

Inputs:

- ✓ Scope Matrix
- ✓ Specific features list

Steps:

1) *Identify the suitability values in the scope matrix.* For each process variation point, take the defined suitability value from Table 3.4 Scope

Matrix that corresponds to the value of the characteristics defined in the new situation

2) *To make the scope evaluation.* To carry out the scope evaluation, consider the following guidelines.
    a. If most of the characteristic of the new situation have an evaluation “Totally adequate TA” suitability, the process asset can support the situation and therefore must be identified with a total assessment of “Totally adequate TA”. Process assets that have a final assessment of “Suitable” should also be identified because they may be subject to change so that they can meet a future specific need. With the total evaluation value of each variation point, it will allow deciding on the process elements that are within and outside the line scope. For this particular case, from Table 3.**6**., task 2 will be part of the process because its final suitability value, for this situation, is TA. Task 1 obtained a final suitability value of I, therefore, it will be out of reach and T3 obtained a final suitability value of S, so T3 may be subject to change so that they can meet a future specific need.
    b. If the suitability final assessment of the process variable elements is equal to N, the decision to be part or not of the process scope must be taken by those in charge of managing the process line. It is necessary to consider that the variation points must take a single alternative either by determination of SpeTion - SPrL or by the decision of the process line managers.
    c. The cross-reference between the situation characteristics and the process assets, through suitability relationships, are a high-level vision of tailoring decisions which must be materialized with the implementation of tailoring rules
    d. Therefore, the process that is suitable for each situation will comprise of the elements selected from their best final suitability assessment.

Characteristics	**Values**	**Variant tasks by variation point or by optionality**		
		Task1:	**Task2:**	**Task3:**
Characteristics A	**Value1**	**I**	**TA**	**S**
	Value2	I	S	N

	Value3	TA	TA	S
Characteristics B	**Value1**	**N**	**S**	**S**
	Value2	N	S	N
	Value3	I	TA	TA
Characteristics C	Value1	I	N	S
	Value2	N	S	N
	Value3	**I**	**TA**	**TI**
Total evaluation		**I**	**TA**	**S**

Table 3.6. Example scope use

Output: Process elements suitable for a specific situation

Customization

The task goal is to establish whether the scope must be resized to support a new situation. This task studies the parts or process assets must be added or removed for facing similar future situations by providing process assets appropriate to that new situation

Objective: To study the realization of adjustments within the scope in case there are not enough process assets to face a specific situation.

Inputs:

- ✓ Scope Matrix
- ✓ specific features List

Steps:

1) Scroll through the scope matrix verifying its “adequate” final valuation status
2) List each of the process assets in "appropriate" state
3) Study and verify each process asset in the “adequate” state in order to identify which parts of its description can be omitted or added so that it can move from the state of its “adequate” suitability to the “totally adequate” state, and thus it is able to support the specific need to a future situation
4) Each change in the suitability status of the process assets must be justified

Output: Updated Scope matrix.

Evaluation

The evaluation is focused on establishing the reasons and/or justifications under which the scope can evolve or not so that a particular situation can be addressed or discarded.

Objective: To determine the reasons why the process line must or must not support a new situation

Inputs:

- ✓ Scope Matrix

Steps:

1) Collect the reasons and/or justifications of each of the process assets that changed their suitability status
2) Evaluate and document the general decision to attend or discard the support of the specific situation

Output: Justification Document

3.4 Synthesis and discussion

This chapter shows SpeTion-SPrL, it was built under the principles of systematicity, integrality, properly conducted and materializable, whose support is determined by the different elements that make it up. SpeTion-SPrL structures two processes that cover two instants related to the analysis and design, and scope use, so that these processes are coherent with the phases of the construction and use of the process lines. The first process supports the scope creation scenario, where it defines the following tasks: Identify projects, identify products, identify situations, to elicit of software process features and suitability determine. The second process defines the following tasks: Specify the situation, Process scope evaluation, Customization and, Evaluation. SpeTion-SPrL, as an integrated approach, adequately try in defining the scope of Software Process Lines, it provided an approach that has processes, tasks, guidelines, and examples which It addresses the two phases of Process Line Engineering, Domain Engineering, and Application Engineering, allowing not only that the materialization of the scope be reflected in the construction of the process lines, but also helps to make adaptation decisions when SPrL is in use.

Chapter 4

SpeTion – SPrL - Validation

4.1 Overview

This chapter shows the different times when SpeTion-SPrL validation was done. The first moment corresponds to empirical validation in the context of the GreenSQA company, which according to the objectives of validation is divided into two parts, part A, which refers to the comparison of scope definition methods CASPER and SCOPE, through the application of these in the practice, in order to determine their advantages and disadvantages. Part B, which refers to validation to determine if the integration of these two methods is feasible. The second moment corresponds to the SpeTion-SPrL application through experimentation in an academic context to determine its ease of use, utility, and reliability. Finally, the last moment corresponds to the confirmatory validation of SpeTion-SPrL through a case study, applied in the same company context of GreenSQA, to validate whether the approach is suitable for defining the scope in a process line.

4.2 Empirical validation – Part A: advantages and disadvantages of applying SCOPE and CASPER

In this part show the real exploration of the SPrL scope definition, in GreenSQA, a software company focused on the testing service through two methods: SCOPE and CASPER. The study was performed as follows: study planning and design, execution and analysis. The details of the development of the empirical study are not shown in detail in this session, but in *annex D, E and K*, it finds all the information.

4.2.1 Planning and design

4.2.1.1 GreenSQA Context

With the aim to know and explore about how to define the software process scope in a real context, developed a preliminary study in GreenSQA, a company dedicated to providing software testing services in the southwest of Colombia. The company was chosen because of the CEO interest in capitalizing the company processes and its associated knowledge. Besides it is having a mature testing process, it is CMMI 3 level, and few organizations in this Colombian region have these characteristics.

4.2.1.2 Goal and research questions

The research goal is to study is to compare two SPrL scoping methods from practical experience in the GreenSQA company context. The research question is: What are the advantages and disadvantages of applying SCOPE and CASPER in practice? This study has one analysis unity, for this case, the testing process family of GreenSQA.

4.2.1.3 Gathering information

To know details about the testing process (reuse/adaptation strategies, process elements, projects, and products) in GreenSQA and its environment (situations characterization, context variables), three techniques for information gathering of qualitative data collection were considered: semi-structured interviews, focus group and surveys. Furthermore, with the aim of identifying the advantages and disadvantages of the methods a workshop was designed. The workshop looking for unveiling the used strategies for defining the scope and identifying the advantages and disadvantages of the methods.

4.2.1.4 Scope determination

For scope determination CASPER and SCOPE methods were applicate. The CASPER application was materialized with development of the process domain engineering that activities include the follows activities *process context analysis*, *process feature analysis* and *scoping*. The SCOPE application was materialized with development of follows activities *product analysis, project analysis, process analysis, attribute prioritization* and *scope determination*.

4.2.2 Discussion and conclusions

After data consolidate, our main results were organized around the advantages and disadvantages related to the scope definition using SCOPE and CASPER methods.

The main SCOPE method advantage is about calculations that quantitatively define the degree of suitability of a process to certain characteristics of both product and project. Calculations serve to determine the required demand of a set process and it is a precise process selection approach based on quantitative criteria, helping to reduce ambiguity through measurement of the suitability. The main SCOPE method disadvantages are: a) it lacks exact guidelines that helps to endeavors to its application b) the domains for specifying the processes demands are limited to projects and products, leaving out particular aspects of the organization c) it only considers processes as a whole for a possible selection (coarse-grain process).

The main advantages from CASPER method are: a) it defines a specific activity for establishing the scope with a set of steps for its elaboration. It has descriptions and elicitation guides of the inputs, such as the context model and process with its variability, that are easy to apply b) It defines adaptation rules at high abstraction level c) It provides freedom of what context means according to the criteria of demand of their processes. d) It defines a variable level of granularity considering the variability of the process and as it could be resolved. The main CASPER method disadvantage is that establish the relationships between the context elements and the variable process elements very limited because it matches them using boolean variables. There may be a conflict to the extent that more than one context variable affects the selection of a process element, that is because different values of context variables may reject the selection of the same process element. However, the inclusion or exclusion of the process elements depend on comparisons of only two possible values, i.e. neither quantitative values nor priorities are used. Table 4.**1**. summarizes the advantages and disadvantages of each method, and Table 4. **2**. shown a summary way the comparison of these methods.

Advantage	Disadvantages

SCOPE	a) Their respective calculations are easy to materialize. b) It defines criteria for the selection of a process by using suitability c) It quantitatively defines the degree of suitability of a process to certain characteristics of both product and project d) Calculations serve to determine the required demand of a process set e) It is a precise process selection approach based on quantitative criteria, helping to reduce ambiguity through measurement of the suitability	a) Its conception is not given as a process b) It does not clearly guide the obtaining of the necessary elements for scoping calculation c) Does not roles and responsibilities that support each of the activities d) The domains for specifying the processes demands are limited to projects and products, leaving out particular aspects of the organization e) It only considers processes as a whole for a possible selection (coarse-grain process).
CASPER	a) It has descriptions and elicitation guides of the method inputs, such as the context model and process with its variability that are easy to apply b) It defines adaptation rules at a high abstraction level c) It provides freedom of what context means according to the criteria of demand of their processes d) It defines a good level of granularity considering the variability of the process and as it could be resolved	a) It establishes limited relationships between the context elements and the process features using only boolean variables i.e. neither quantitative values nor priorities are used. Also, these relationships are hard to identify. b) There may be a conflict to the extent that more than one context variable affects the selection of a process element, that is because different values of context variables may reject the selection of the same process element c) It is not clear as adaptation rules to be gathered and where these came from

Table 4.1. Advantage and disadvantages summary of methods

	Comparison elements	SCOPE	CASPER
a)	Its description defines some activities or steps for scoping definition	Activities set	Activities set
b)	Its description defines some roles or skills for scoping definition	No	Yes
c)	Identification of the range of characteristics	Yes, but only three dimensions	Yes, but only consider actual context elements
d)	Determine the situations where the process will be used and what process elements (common and variables) will be required in each situation	Yes, but does no determine what process elements will be required in each situation.	Yes, but only consider actual situations
e)	It has guidelines for supporting	No	Yes
f)	It defines tailoring decision	No	Yes
g)	Uses support tools or technology to support the scoping definition	Yes	Yes
h)	Uses criteria for helping in the selecting assets or process	Yes, suitability values from 0 to 1	Yes, three-value variables

Table 4. 2. Summary of the methods comparison

According to the identified weaknesses, both methods have limitations in practice. They have different degrees of granularity, high level (process) for SCOPE and low level for CASPER (process assets), which give support to different but complementary aspects. Although the approaches provide some elements that help in their application, it is not enough to clear how systematically conducting them, particularly the information gathering from explicit and implicit sources. Due to the different levels of granularity of the two approaches, the scope concept is not equally interpreted, where CASPER tries to solve the selection of the current assets it does not consider future assets and its probabilities of being used, instead of it considers variability and its resolution as part of the scope. On the other hand, SCOPE, by calculating the level of adequacy of the process according to demands, tries to help in the selection of current, planned and future processes, but does not take in an account about the identification of variability and its possible resolution. Former findings allow us to conclude that lack of an acceptable definition about what the scope of a SPrL is, how it is represented and defined.

4.3 Empirical validation – Part B: an integrated approach between SCOPE and CASPER

This part presents an empirical validation of the first version of the hybrid method proposed in this thesis. This evaluation is based on the entire infrastructure defined in the previous section (Part A) reported in Ruiz et. al (Ruiz & Hurtado, 2019), where the main goal was to explore, through an empirical study, the advantages and disadvantages of scope determination in an industrial context by using of SCOPE and CASPER methods. The goal of this empirical study is to inquire about the feasibility of the integration of SCOPE and CASPER methods in a unique approach.

4.3.1 Planning and Design

4.3.1.1 GreenSQA context

The empirical study was conducted in GreenSQA company, the company actively belongs to the Colombian software industry, located in Cali city, dedicated to quality assurance and software testing. GreenSQA is a company with 15 years of experience in the field of software testing with more 15.000 projects developed successfully. It has a team of highly qualified professionals with a flexible approach aimed at solving software industry problems. Specifically, for the study development, 6 employees helped us with the empirical study development and with the solution of necessary surveys, the employees were expert engineers in the company's testing strategies, they thoroughly knew the projects carried out and the company's environment, therefore, they were the right people to support the study development, considering the importance of having people with experience so that the results were the most indicated.

4.3.1.2 Goal and research questions

The empirical study goal was to inquire about the feasibility[4] of the integration of a hybrid approach for scope determination, from the perspective of the expert engineers in test strategies of GreenSQA company. For this, the following research question was stated: *How feasibility is this integration?* This study has one analysis unity, that is the GreenSQA testing process. The organization selection was largely since it offered good availability and interest in capitalizing on its process.

4.3.1.3 Hypothesis

Considering the study objectives, it is intended to evaluate the following hypothesis:

a) An integrated approach between SCOPE and CASPER is feasible for scoping determination

In order to refine the previous hypothesis, the following specific hypotheses were raised, see Table 4.**3**.

Hypothesis		**Variables**
Feasibility	H.1. The users perceive that the tasks or steps based on CASPER and SCOPE for scoping definition are available on SpeTion-SPrL	*Perceived availability*: Refers to the perceived availability degree per person in the scope definition. This variable represents a tasks availability perceptual judgment, or the steps defined in SpeTion-SPrL.
	H.2. The users perceive that they achieve a scope definition artifact based on CASPER and SCOPE	*Materialized scope:* the perceived scope materialization degree per person in its determination in a process line. This variable represents a scope materialization perceptual judgment for a software processes line using SpeTion-SPrL.

Table 4.3. Study Hypothesis

4.3.1.4 Design

---

[4] The feasibility refers to the availability of the resources necessary to carry out the stated objectives

Table 4.**4.** shows the activities designed for the study development and also specifies the activities duration and the support instruments that were used for its development.

Study activities	Planned duration	Support instruments
Activity 1: Socialize and contextualize the study	30 min	Presentation of introduction to study
Activity 2: Socialize the CASPER and SCOPE execution results	1 hours	Results presentation
Activity 3: Apply the scope definition proposal in a guided manner	2 hours	Scope definition proposal document
Activity 4: Solve the questionnaire	15 min	Surveys
Total time: 3 hours 55 min		

Table 4.4. Study activities summary

4.3.2 Execution

a) Activity 1: For the development of this activity in the first part, an oral presentation was made in order to socialize and contextualize the participants with the study. In addition, in the final part of this activity, the study activities schedule presentation was made, Table 3, from which questions about some applied concepts arose.

b) Activity 2: It is necessary to remember that this study is the continuation of the empirical study shown in (Ruiz & Hurtado, 2019) so that, this activity focused on socializing and presenting the results and inputs, required for the CASPER and SCOPE execution, in the scope definition obtained in previous sessions. The main objective of this activity was to get the participants to remember and know the data and their sources so that they could later use them in the execution of the next activity.

c) Activity 3: As an initial part of this activity, it was oriented in a general way, what was our proposal for the scope definition. In the second part, the study participants applied our proposal, in a guided way. As a result of this activity, the scope matrix was obtained which condenses the process features suitability.

d) Activity 4: In this last activity, the participants answered a survey that allowed us to evaluate the feasibility of an integrated approach.

4.3.3 Aanalysis and results

For the hypothesis validation, results were obtained by conducting a survey (See the Annex *L*: Evaluation of SpeTion - SPrL use – Empirical study) by employees at the end of the activities (activity 4) which was taken as a data source.

The analysis was carried out by selecting the results of the survey solved by the six employees who participated in the study. The survey responses were based on the Linkert scale, which is a form of measurement that allows to assess attitudes and know the conformity degree on a statement set. The measurement scale of the survey responses was defined as follows: value 1 for the *totally disagree* option, value 2 for the *disagree* option, value 3 for the *neutral* option (neither agree nor disagree), value 4 for the *agreement* option and value 5 for the option *totally in agreement*. Of the hypotheses initially drawn, the following null hypotheses were raised:

- H.1_0, π_1 <= 60%, where $\pi 1$ is the perception percentage that evaluates the availability of tasks or steps based on CASPER and SCOPE for scoping definition
- H.2_0, π_2 <= 60%, where π_2 is the perception percentage that evaluates that achieve a scope definition artifact based on CASPER and SCOPE

From the null hypotheses the following alternative hypotheses were obtained:

- H.1, π_1 > 60%, where $\pi 1$ is the perception percentage that evaluates the availability of tasks or steps based on CASPER and SCOPE for scoping definition
- H.2, π_2 > 60%, where π_2 is the perception percentage that evaluates that achieve a scope definition artifact based on CASPER and SCOPE

To validate the hypotheses raised, the participants resolved the designed survey, the values obtained per participant for each of the questions are presented below in Table 4.**5.**

	ES1	**ES2**	**ES3**	**ES4**	**ES5**	**ES6**
Participant 1	3	4	4	4	4	4
Participant 2	2	3	2	3	4	4

Participant 3	3	3	4	2	3	3
Participant 4	3	3	3	4	4	4
Participant 5	4	4	3	4	4	4
Participant 6	4	4	4	4	3	3

Table 4.5. Values for each question

To validate the H1 hypothesis, the following six assertions were validated:

- ES1 - SpeTion-SPrL approach has the necessary tasks available for defining the process line scope.
- ES2 - SpeTion-SPrL approach has the necessary steps available to support the process line scope definition.
- ES3 - The tasks defined in the SpeTion-SPrL approach allow defining the process line scope.
- ES4 - The steps defined in the SpeTion-SPrL approach support the process line scope definition.
- ES5 - With the SpeTion-SPrL approach use, it is possible to define the process line scope.
- ES6 - The using result the SpeTion-SPrL approach is an artifact where the process line scope materializes.

The results obtained of validated assertions are shown in the following figures Table 4.**6**, Table 4.**7**, Table 4.**8**, Table 4.**9**, Table 4.**10**, Table 4.**11**.:

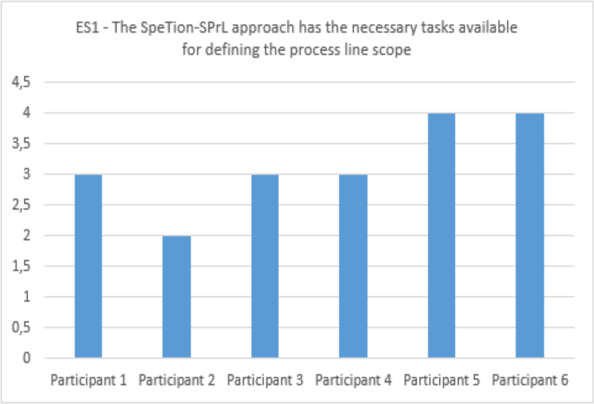


Table 4.6. ES1 results

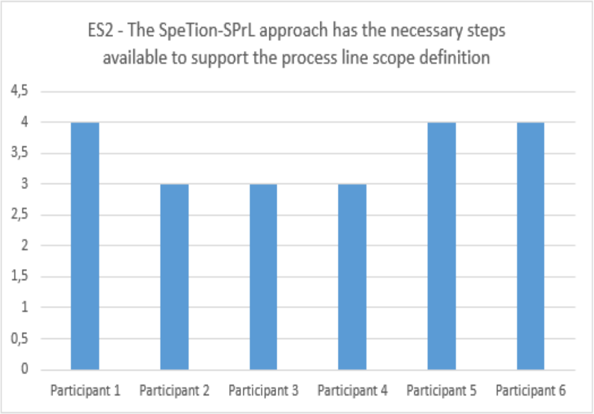


Table 4.7. ES2 Results

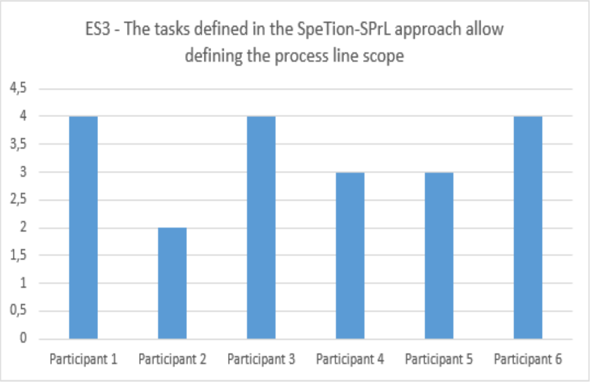


Table 4.8. ES3 results

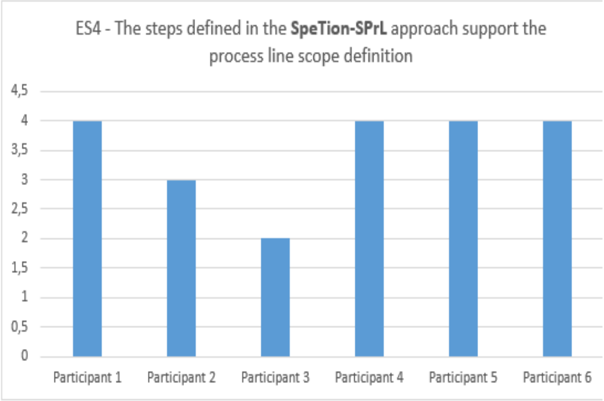


Table 4.9. ES4 results

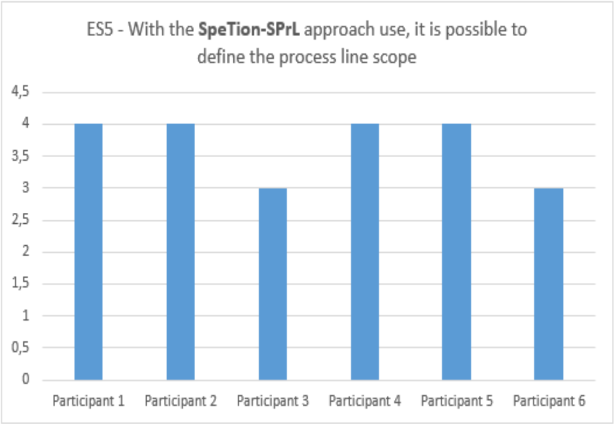


Table 4.10. ES5 results

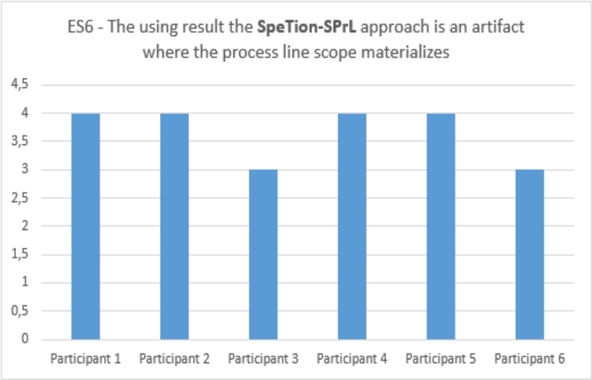


Table 4.11. ES6 results

Of the values obtained we can say (considering that for answers of "agreement" and "totally in agreement", the response is taken as positive):

- Assertion ES1, the perception percentage of respondents is 33,3%, that determined that the SpeTion-SPrL approach has necessary tasks available for defining the process line scope.
- Assertion ES2, the perception percentage of respondents is 50%, that determined that the SpeTion-SPrL approach has the necessary steps available to support the process line scope definition.
- Assertion ES3, the perception percentage of respondents is 50%, that determined that the tasks defined in the SpeTion-SPrL approach allow defining the process line scope.

- Assertion ES4, the perception percentage of respondents is 66,7%, that determined that the steps defined in the SpeTion-SPrL approach support the process line scope definition.

To validate the H2 hypothesis, the following two questions were considered:

- Assertion ES5, the perception percentage of respondents is 66,7%, that determined that with the SpeTion-SPrL approach use, it is possible to define the process line scope.
- Assertion ES6, the perception percentage of respondents is 66,7%, that determined that with the using result the SpeTion-SPrL approach is an artifact where the process line scope materializes.

According to the results, it can be inferred that:

- With the assertions with lower percentages than 60%, H.1 can be rejected. In this way, it can be said that the tasks or steps based on CASPER and SCOPE for scoping definition aren't available.
- H.2 can be accepted. In this way, it can be said that with SpeTion-SPrL achieve a scope definition artifact based on CASPER and SCOPE.

4.3.3.1 Conclusions

This study validates the feasibility of integrating the SpeTion-SPrL approach, for this H1 and H2 were defined. Hypothesis H1 make reference to the user's perception that the tasks or steps based on CASPER and SCOPE for scoping definition are available on SpeTion-SPrL, and the hypothesis H2 make reference to the user's perception that they achieve a scope definition artifact based on CASPER and SCOPE. According to the validation of these two hypotheses, it can be said that the participants managed to perceive that SpeTion-SPrL does not have all the availability of tasks and steps to define the scope, however, it was materialized thanks to the fact that the participants had good experience both in the company's testing processes and in an ad hoc adaptation strategy. According to the above, it can be said that the integration between SCOPE and CASPER is moderately feasible for scoping determination taking into account that a scope artifact is reached but the integrated approach does not fully have the necessary tasks and steps for its definition.

4.3.3.2 Threads of validity

Construct validity: Participants and researchers do not interpret the questions defined in the survey in the same way. In order to minimize this effect, the instruments were previously validated between researchers.

Internal validity: a threat of internal validity is the participants' number involved in study development by the company. Although the selection of the members was low, they had the necessary experience to support the study development in an adequate way.

External validity: For studying develop the organization must have a well-defined process, thus is, a process maturity, which will most likely not be the case of many Colombian software development companies.

4.4 Academic Experiment

4.4.1 Context experiment

As a result of this experiment there was an article (See Annex *K*).

The experiment was conducted in a university environment in which students from the following universities participated: Universidad de la Matanza-UM (Argentina), Universidad Nacional de la Plata -UP (Argentina) and the Universidad del Cauca- UC (Colombia). UM students were in the last semester of the systems engineering program. This university has the characteristic of having a large percentage of students who are working in different knowledge areas. Specifically, for the experiment development, there were 45 students involved with companies related to software development. For its part in the UP, 20 students participated who were in the third year of the systems engineering program; and finally, the UC participants were 5 students of the master's program in computer science, specifically of the process engineering course. Details of the experimentation participants are shown in Table 4.12.

Institution	Participants	Characteristics
UM	45	Senior systems engineering students working in companies related to software development.
UP	20	Third-year systems engineering students
UC	5	Master's students in computing
Population	70	

Table 4.12. Population characteristics summary

4.4.2 Design experiment

Table 4.**13**. summarizes the activities designed for the experiment development and specifies the activities duration and the support instruments that were used for its development.

Experimentation activities	Planned duration	Support instruments
Activity 1: Socialize and contextualize the experiment	20 min	Presentation of the introduction to the experiment and conceptual elements
Activity 2: Submit the scope definition proposal	20 min	Document with the approach description and presentation
Break	10	None
Activity 3: Apply the scope definition proposal	1 hora 30 min	Task document, scope definition guide document, scenarios and result templates for approach execution
Activity 4: Solve the questionnaire	15 min	Surveys
Total time: 2 hours 45 min		

Table 4.13. Experimentation activities summary

It is important to clarify that, as part of the study design, the support instruments were subjected to several revisions, so that they met the needs of the experiment and also to improve their description, in which two members of the IDIS research group of the Universidad del Cauca and a member of the GIS group of the Universidad de la Matanza participated.

The experiment design was carried out in two parts: qualitative design and quantitative design. The qualitative analysis was made for the hypotheses validation H1, H2, H3 and H4, for which the results obtained with the completion of a survey (activity 4) were taken as a source of data. In this way, for the qualitative analysis the sample calculation is given by the following equation:

$$n = \frac{K^2 * p * q * N}{e^2(N-1) + K^2 * p * q} \quad (1)$$

Where K is the confidence level $K = 1.65$, e is the sampling error $e = 0.10$, p proportion of individuals who have the study characteristic in the population. This data is generally unknown, and it is usually assumed that $p = 0.50$. On the other hand, q is the proportion of individuals who do not have that characteristic, that is, it is 1- p and *N* is population size *N* = 70. Where the sample size $n = 35$. The selection of the 35 surveys was made randomly.

The quantitative analysis was done using the results embodied in the templates provided by our proposal to define the scope (activate 3). Since the method was executed through groups of five participants, a total of 14 groups were organized in such a way that they had the same number of method execution templates. Equation (1) was used to calculate the sample for this case but considering that the value of *N* = 14, resulting in a sample size $n = 12$. The selection of the 12 templates was made randomly.

4.4.3 Objective and hypothesis

To explore the scope definition proposal in order to evaluate its ease of use, utility, and reliability from the perspective of the process engineers' group in the Universities student's context of Nacional de la Matanza, Universidad de la Plata, and Universidad del Cauca. Considering the objective of the experiment, the following hypotheses were evaluated, see Table 4.**14**.

Hypothesis	**Variables**	
H.1.1 Proposal users understand the instructions and guidelines of the approach	*Ease of understanding:* it is the degree of ease with which a person can understand the proposal. This variable represents a perceptual judgment of the effort required to understand our proposal to define the scope	Ease of use
H.1.2 Proposal users understand the support examples provided by the approach		
H.2 Proposal users perceive that the approach has enough support information to guide its application	*Ease of application:* the degree of ease with which a person can apply the proposal to define the scope in a process line. This variable represents a perceptual judgment of the effort required to apply the proposed scope definition	
H.3.1 Proposal users perceive that the approach is useful for building a scope that helps in the process line delimitation	*Perceived utility:* it is the utility degree perceived per person in defining the scope of a process line. This variable represents	Utility

H.3.2 Proposal users perceive that the approach is organized and consistent	a useful perceptual judgment of the proposal to define the scope	
H.4.1 Proposal users perceive that the executing result the approach is useful for process line development	*Results usefulness:* It is the perceived utility degree of the results per person in scope definition of a process line. This variable represents a utility perceptual judgment of the results of the scope definition proposed	
H.5.1 The suitability results expressed by the proposal users during its execution are like each other	*Reliability of the approach:* It is the degree to which the proposal provides similar results in the suitability identification of the process variable elements. This variable represents a comparison of the suitability values, that are the results of the approach application	Reliability

Table 4.14. Experiment hypothesis

4.4.4 Execution

Although the development of this experimentation was carried out in three different universities and at different times, the activities were executed in the same way in each of them, always conserving the same material and the experiment structure. For this reason, it will be described in a general way how the activities execution in each of the universities was done. Table 4.**15**. shows the time invested in each of the activities.

Activities	UM	UP	UC	Estimated time
Activity 1	30	30	20	20
Activity 2	35	25	25	20
Break	15			10
Activity 3	1:45	1:53	2:40	1:30
Activity 4	10	12	15	15
Total	3h:35m	3h:15m	3h:55m	2h:45m

Table 4.15. Time invested in each activity

- Activity 1: This first part aimed to socialize and contextualize in a general way what the experiment was. To carry out this activity, an oral presentation was made in order to introduce and inform the participants of the experiment, about the activities that would be developed, in addition to making known and clarifying some concepts used in it.

- Activity 2: This activity was guided by an oral presentation of the scope definition proposal so that the participants had a method overview that they would subsequently apply.
- Activity 3: As an initial part of the development of this activity, the participants were guided in the application of our proposal and in the support material used. In the second part of this activity, the participants of the experiment applied our proposal in order to solve the process line scope definition (See Annex *M*, where is the task to be performed (Section 1: Task), the guide they should follow (Section 2: SPETION-SPrL-Guideline) and the scenario on which they should work (Section 3: Scenario)). As a result of this activity, the participants filled out several templates (See Annex *M*, Section 4: Templates), among them the one that condenses the process features suitability which was used to test the hypothesis H.5.
- Activity 4: In this last activity, the participants answered a survey, which allowed us to evaluate the ease of use and usefulness of the approach (See Annex *M*, Section 5: Survey, Section 6: Questions).

4.4.5 Qualitative analysis and results

The qualitative analysis was done by selecting 35 surveys according to our sample value. The 35 surveys selection was made at random. The survey responses were based on the Linkert scale, which is a form of measurement that allows to assess attitudes and know the conformity degree on a set of statements. The measurement scale of the survey responses was defined as follows: value 1 for the *totally disagree* option, value 2 for the *disagree* option, value 3 for the *neutral* option (neither agree nor disagree), value 4 for the *agreement* option and value 5 for the option *totally in agreement*. Of the hypotheses initially drawn, the following null hypotheses were raised:

- $H.1.1_{0,}$ $\boldsymbol{\pi}_1 <= 60\%$, where $\boldsymbol{\pi}_1$ is the perception percentage that evaluates the ease of understanding of the instructions and approach guidelines
- $H.1.2_{0,}$ $\boldsymbol{\pi}_2 <= 60\%$, where $\boldsymbol{\pi}_2$ is the perception percentage that evaluates the ease of understanding of the approach examples
- $H.2.1_{0,}$ $\boldsymbol{\pi}_3 <= 60\%$, where $\boldsymbol{\pi}_3$ is the perception percentage that evaluates that the approach has enough information for its application

- H.3.1$_0$, **π_4** <= 60%, where **π_4** is the measurement that evaluates the approach utility perception to build a scope that guides and delimits the process line
- H.3.2$_0$ **π_6** <= 60%, where **π_6** is the measurement that evaluates the perception that the approach is organized and consistent
- H.4.1$_0$, **π_5** <= 60%, where **π_5** is the measurement that evaluates the utility perception of the result obtained for a process line development

From the null hypotheses the following alternative hypotheses were obtained:

- H.1.1, **π_1** >60%, where **π_1** is the perception percentage that evaluates the ease of understanding of the instructions and approach guidelines
- H.1.2, **π_2** >60%, where **π_2** is the perception percentage that evaluates the ease of understanding of the approach examples
- H.2.1, **π_3** >60%, where **π_3** is the perception percentage that evaluates that the approach has enough information for its application
- H.3.1 **π_4** > 60%, where **π_4** is the measurement that evaluates the approach utility perception to build a scope that guides and delimits the process line
- H.3.2. **π_5** > 60%, where **π_5** is the measurement that evaluates the perception that the approach is organized and consistent
- H.4.1. **π_6** > 60%, where **π_6** is the measurement that evaluates the utility perception of the result obtained for a process line development

To statistically validate our hypotheses, we use a test statistic that is calculated from the data obtained by the sample in the survey completion. With the test statistic calculation, these results are compared with what is expected in the null hypotheses. Equation (2) was used to calculate the test statistic.

$$z = \frac{\hat{p} - \pi}{\sqrt[2]{\frac{\pi(1-\pi)}{n}}} \quad (2)$$

Where $\hat{p}$ estimated is calculated by the favorable cases number, survey responses that are above 3 according to Linkert, on the number of total cases, π is the value considered to validate the hypotheses. To find the critical values a significance α=0.1, was used, to find the critical value is done using the normal distribution table where $Z_{\alpha}=Z_{0.1}=1.28$, called critical value *vc* in the hypothesis tests. Table 4.**16.** summarizes the calculations made for the hypothesis tests.

Hypothesis	**Z (statistical)**	**Vc**	**Hypothesis testing**
$H.1.1_0$	-1.035	1.28	It is not rejected
$H.1.2_0$	-1.035	1.28	It is not rejected
$H.2_0$	1.380	1.28	Is rejected
$H.3_0$	2	1.28	Is rejected
$H.3.2_0$	1.370	1.28	Is rejected
$H.4.1_0$	1.723	1.28	Is rejected

Table 4.16. Summary of calculations for hypothesis validation.

According to the results of Table 4.**16.**, it can be inferred that:

- With the data provided by the sample, there is insufficient evidence to reject $H1.1_0$, therefore, it is possible that the participants failed to perceive a satisfactory understanding of the instructions and guidelines contemplated by approach.
- With the data provided by the sample, there is not enough evidence to reject $H.1.2_0$, therefore, it is possible that the participants failed to satisfactorily understand the support examples provided by the approach.
- $H.2_0$ is rejected and, therefore, the alternative hypothesis H.2. is accepted. In this way, users of our proposal perceive that the approach has enough support information to guide its application.
- $H.3.1_0$ is rejected and, therefore, the alternative hypothesis H.3.1 is accepted. In this way, the users of our approach perceive that the approach is useful for building a scope that helps in the process line delimitation.
- $H.3.2_0$ is rejected and, therefore, the alternative hypothesis H.3.2 is accepted. In this way, the users of our approach perceive that the approach is organized and consistent.
- $H.4.1_0$ is rejected and, therefore, the alternative hypothesis H.4.1 is accepted. In this way, users of our approach perceive that the result of executing the approach is useful for the process line development.

4.4.6 Quantitative analysis and results

The analysis in this section was based on the suitability data comparison provided by the groups during the execution of our approach, in order to establish whether they are statistically similar. The hypotheses for each of the process variable activities are shown below:

Null hypothesis:

- $H.5.1_0$ the suitability values set by each group for the variable activity: *plan the integration* are similar
- $H.5.2_0$ the suitability values set by each group for the variable activity: *peer programming* are similar
- $H.5.3_0$ the suitability values set by each group for the variable activity: *individual programming* are similar
- $H.5.4_0$ the suitability values set by each group for the variable activity: *code generation* are similar
- $H.5.5_0$ the suitability values set by each group for the variable activity: *integrate each subsystem* are similar

Alternative hypotheses:

- H.5.1 the suitability values set by each group for the variable activity: *plan the integration* are not similar
- H.5.2 the suitability values set by each group for the variable activity: *peer programming* are not similar
- H.5.3 the suitability values set by each group for the variable activity: *individual programming* are not similar
- H.5.4 the suitability values set by each group for the variable activity: *code generation* are not similar
- H.5.5 the suitability values set by each group for the variable activity: *code generation* are not similar

The Chi-Square test statistic (χ2) was used to make the analysis using the following equation:

$$\chi 2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(Aij - Eij)^2}{Eij} \quad (3)$$

Where, Aij = is the value observed in the ith row, jth column, Eij = is the expected value in the ith row, jth column, r = rows number and c = columns number. Table 4.**17**. shows the observed and expected values in the specific case of the variable activity *plan the integration*.

Group	1	2	3	4	5	6	7	8	9	10	11	12
Suitability per group	0,58	0,58	0,62	0,58	0,63	0,66	0,38	0,60	0,50	0,62	0,52	0,60
Observed values	3	3	3	3	3	3	2	3	3	3	3	3
Expected value	2	2	2	2	2	2	3	2	2	2	2	2

Table 4.17. Observed values of suitability by group to plan integration

The first row refers to each of the groups that participated in the experimentation. The second row corresponds to the suitability value defined by each of the groups for the task *Plan the Integration*. The third row refers to how many of the group members agreed with the suitability value defined in row 2, and the expected value is reflected in the fourth row. The χ2 calculation was done using the highlighted rows data in table 6 using equation (3), where χ2 = 6.39. The degrees of freedom for the χ2 test is given by (rows -1) * (columns -1), which for this case is 11. The critical value is established using the distribution table of χ2 with 11 degrees of freedom and significance α = 0.5, therefore, *vc* = 19.67. In this way the values of χ2 that are greater than the *vc* can reject the null hypothesis otherwise it must be accepted. The calculation of the χ2 test for the "Plan integration" task is 6.39, therefore, H1.0 is accepted. Table 4.**18**. shows the result of χ2 for all the process variable activities whose calculation was made in a similar way to that shown for the activity to *plan the integration* but considering the appropriate suitability values for each variable activity.

Variable activities of the process	**χ2**	***Vc***	**Hypothesis testing**
Plan the integration	6.39	19.67	Acceptance of $H.5.1_0$
Peer programming	14.04	19.67	Acceptance of $H.5.2_0$
Individual programming	5.48	19.67	Acceptance of $H.5.3_0$
Code generation	13.86	19.67	Acceptance of $H.5.4_0$
Integrate each subsystem	15.12	19.67	Acceptance of $H.5.5_0$

Table 4.18. Summary of calculations for hypothesis validation using χ2

According to the results of Table 4.**18.**, it can be inferred that:

- $H.5.1_0$ can be accepted. In this way, it can be said that the suitability values established by each group for the variable activity *plan the integration* are statistically similar
- $H.5.2_0$ can be accepted. In this way, it can be said that the suitability values established by each group for the variable activity *Peer programming* are statistically similar
- $H.5.3_0$ can be accepted. In this way, it can be said that the suitability values established by each group for the variable activity *Individual programming* are statistically similar
- $H.5.4_0$ can be accepted. In this way, it can be said that the suitability values established by each group for the variable activity *Code generation* are statistically similar
- $H.5.5_0$ can be accepted. In this way, it can be said that the suitability values established by each group for the variable activity *Integrate each subsystem* are statistically similar

The results obtained by each university are found in Annex *M*, Section 7: Results obtained.

4.4.7 Conclusions

This work validates the ease of use, utility, and reliability of an approach to scope definition in an SPrL. For this, an experiment was developed with students from three Latin American universities. According to the hypotheses validation H.1.1 and H1.2 regarding the understanding of our proposal, it can be said that the participants failed to have a complete understanding of the instructions, guidelines, and examples that are part of the approach, in such a way, this allows us to identify future improvements in the description and elements that make up our approach. In accordance with the validation of H.2. it can be concluded that the participants perceived that the approach has enough information for its application, but it cannot be ensured that this information is adequate to achieve a satisfactory understanding of the information. Although H.1.1 and H1.2 are inconclusive, participants perceived that there is a utility in the approach to support the scope delimitation of an SPrL, through an organized and consistent approach, in accordance with the H.3.1 and H3.2 validation. In addition, the H.4.1 validation allows us to say that the results obtained when executing the approach are also utility. Considering that our proposal is perceived as utility, organized, consistent and has enough information, we can infer that there is a conceptual complexity that does not allow an adequate understanding of all the work involved in scope defining. Regarding the approach reliability, it can be said that the resulting suitability values are statistically similar. These values were achieved systematically and independently of the group for each of the process elements, in this way it can be said that the participants arrived at the same solution, therefore, hypotheses H.5.10, H.5.20, H.5.30, H.5.40, H.5.50, are accepted, so that, in this context, the approach can be considered systematic and can, therefore, help determine the inclusion or exclusion of process features for a specific situation.

4.4.8 Threads of validity

Construct validity: to minimize the subjectivity in the instruments supporting the collection of study information, during the study planning the instruments underwent several validations in which two members of the IDIS research group of the Universidad del Cauca and one of the GIS group from the Universidad de la Matanza participated. Another threat identified in this type is the incorporation and management of new conceptual and language elements in the study development. In order to reduce this threat as part of the study, an initial activity was dedicated in which participants were socialized and contextualized in details of it to achieve a concepts unification and language.

Internal validity: the method execution results may be conditioned due to the use of the same problem and types of participants in the scope definition. To mitigate this threat, the study considered that the problem resolution was made through the various participating groups. Another threat of validity of this type may be the time invested for the study execution, being long sessions, participants in the final stages of the experiment can perceive fatigue which can influence the results. To try to mitigate this threat in the middle of experimentation, participants took a break without communication between them.

External validity: due to the experiment was conducted in an academic environment, the results cannot be generalized to a real industry context. To mitigate this threat, participants with experience, in software development, in process engineering, and with no experience were considered to try to emulate a real context one of the industry. In addition, to carry out the experiment, all the necessary supplies were provided to execute the method which does not happen in the industry.

4.5 GreenSQA Case study

4.5.1 Context

The case study was conducted in GreenSQA company, the company actively belongs to the Colombian software industry, located in Cali city, dedicated to quality assurance and software testing. GreenSQA is a company with 15 years of experience in the field of software testing with more 15.000 projects developed successfully. It has a team of highly qualified professionals with a flexible approach aimed at solving software industry problems. Specifically, for the study development, 3 employees helped us with the case study development and with the solution of necessary surveys, the employees were project managers, expert engineers in the company's testing strategies, they thoroughly knew the projects carried out and the company's environment, therefore, they were the right people to support the study development, considering the importance of having people with experience so that the results were the most indicated.

4.5.2 Goal and research questions

To analyze the SpeTion-SPrl approach with the purpose of evaluating its suitability from the perspective of process engineers team in the context of process line definition in the Green SQA company. Table 4.**19**. shows some details related to the objective of this case study.

Objective OB1: SpeTion-SPrl suitability	
Analyze	The SpeTion-SPrL approach
With the purpose of	Evaluate it
With respect to	Suitability
From the viewpoint of	Process engineering team
In the context of	Process line definition in the Green SQA company
Questions associated with the objective	
P1	Do SpeTion-SPrL processes adequately guide the scope achievement?
P1	Are the artifacts made according to the steps established by SpeTion-SPrL?
P2	How complete are specified the artifacts with respect to those presented by SpeTion-SPrL?

P3	How ambiguous is the specified scope to size the process line construction?

Table 4.19. Study case objective

4.5.3 Hypothesis

Considering the case study objective, it is intended to evaluate the following hypothesis:

- The result of SpeTion-SPrl is suitable for determining the process line scope of GreenSQA

In order to refine the previous hypothesis, the following specific hypothesis were proposed, see Table 4.11. Furthermore, for each hypothesis were defined metrics, Table 4.**20.**, Table 4. **21.**, Table 4.**22.**, Table 4. **23.**, Table 4.**24.**, shows it.

Hypothesis		
Systematicity		H.1. SpeTion-SPrL processes are properly conducted
Suitable		H.2 SpeTion-SPrL results are useful
	Well specified	H.3 The SpeTion-SPrL results are clear
		H.4 SpeTion-SPrL results are tangible

Table 4.20. Specific hypotheses

<table>
<tr><th colspan="2">Metric OB1.M1: Systematic</th></tr>
<tr><td>Indicator</td><td>Systematic</td></tr>
<tr><td>Definition</td><td>It refers to the orderly way in which the SpeTion-SPrL approach can be advanced according to its definition. This indicator represents a steps percentage taken in order during the SpeTion-SPrL regarding the steps defined in the approach</td></tr>
<tr><td>Objective</td><td>Determine if the approach was followed in practice according to its specification</td></tr>
<tr><td>Metrics</td><td><table><tr><th>Metric</th><th>Description</th></tr><tr><td>TS: Total steps</td><td>SpeTion total approach steps</td></tr><tr><td>SS: Skipped steps</td><td>Steps omitted by practitioners</td></tr></table>Systematic = (1 -SS /TS) * 100</td></tr>
<tr><td>Periodicity</td><td>1</td></tr>
<tr><td>Analysis procedure</td><td>Percent calculation at higher percentage value, greater systematicity with the approach application</td></tr>
<tr><td>Representation form</td><td>Numerical</td></tr>
</table>

Instruments	Identification rubric for messy steps. SpeTion-SPrL guide, tasks list used in the case study
Responsibilities	Pablo Ruiz

Table 4. 21. Systematicity in the application

<table>
<tr><th colspan="2">Metric OB1.M2: Results utility</th></tr>
<tr><td>Indicator</td><td>Utility</td></tr>
<tr><td>Definition</td><td>It refers to how the scope resulting from executing SpeTion-SPrL supports future projects in the same domain. This indicator represents a comparison between the elements defined in the scope and the elements that will really support the creation of future projects</td></tr>
<tr><td>Objective</td><td>Determine if the elements defined in the scope will support the future projects creation</td></tr>
<tr><td>Metrics</td><td><table>
<tr><th>Metric</th><th>Description</th></tr>
<tr><td>SE: Scope elements</td><td>Total elements defined in scope</td></tr>
<tr><td>SEWU: Scope elements that will be used</td><td>Total elements of the scope that will be used in future projects</td></tr>
</table>
Utility = (1 - (SEWU/SE)) * 100</td></tr>
<tr><td>Periodicity</td><td>1</td></tr>
<tr><td>Analysis procedure</td><td>Percent calculation at lower percentage value, greater scope utility</td></tr>
<tr><td>Representation form</td><td>Numerical</td></tr>
<tr><td>Instruments</td><td>Identification rubric of scope elements useful in future projects. Scope matrix</td></tr>
<tr><td>Responsibilities</td><td>Pablo Ruiz</td></tr>
</table>

Table 4.22. Results utility

<table>
<tr><th colspan="2">Metric OB1.M3: Clarity</th></tr>
<tr><td>Indicator</td><td>Clarity</td></tr>
<tr><td>Definition</td><td>It refers to how the scope resulting from running SpeTion-SPrL provides clarity for its interpretation. This indicator represents the inconsistencies identified in the resulting scope</td></tr>
<tr><td>Objective</td><td>Determine if there are inconsistencies in the elements defined in the scope</td></tr>
<tr><td>Metrics</td><td><table>
<tr><th>Metric</th><th>Description</th></tr>
<tr><td>SE: Scope elements</td><td>Total elements defined in scope</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>ISE: Inconsistent scope elements</td><td>Total elements of scope that are considered inconsistent</td></tr></table>
Clarity = (1 - (ISE/SE)) * 100</td></tr>
<tr><td>Periodicity</td><td>1</td></tr>
<tr><td>Analysis procedure</td><td>Percent calculation at higher percentage value, greater clarity in scope</td></tr>
<tr><td>Representation form</td><td>Numerical</td></tr>
<tr><td>Instruments</td><td>Identification rubric of inconsistent scope elements. Scope matrix</td></tr>
<tr><td>Responsibilities</td><td>Pablo Ruiz</td></tr>
</table>

Table 4. 23. Clarity

<table>
<tr><th colspan="2">Metric OB1.M4: Tangible</th></tr>
<tr><td>Indicator</td><td>Tangible</td></tr>
<tr><td>Definition</td><td>The scope, which is the result of executing SpeTion-SPrL is materialized. This Indicator represents the gradually outputs materialized of the tasks for making up the scope</td></tr>
<tr><td>Objective</td><td>To determine if the approach materializes the scope gradually using the tasks defined in SpeTion execution</td></tr>
<tr><td>Metrics</td><td><table><tr><th>Metric</th><th>Description</th></tr><tr><td>TA: Total Task</td><td>Total task of the approach</td></tr><tr><td>TO: Total Outputs</td><td>Total generated output by the approach</td></tr></table>
Tangible = (1 – (TO /TA)) * 100</td></tr>
<tr><td>Periodicity</td><td>1</td></tr>
<tr><td>Analysis procedure</td><td>Percent calculation at zero percentage value, the scope is materialized, to different percentage values the scope is not materialized</td></tr>
<tr><td>Representation form</td><td>Numerical</td></tr>
<tr><td>Instruments</td><td>Identification rubric for task materialized. Taks defined in SpeTion-SPrL</td></tr>
<tr><td>Responsibilities</td><td>Pablo Ruiz</td></tr>
</table>

Table 4.24. Tangible

4.5.4 Design

Table 4. **25.** shows the activities designed for the case study development and specifies its duration and the support instruments that will be used for its development.

Experimentation activities	**Planned duration**	**Support instruments**
Activity 1: Socialize and contextualize the case study	30 min	Presentation of introduction to experiment and conceptual elements
Activity 2: Submit SpeTion-SPrL	30 min	Document with the approach description and presentation
Activity 3: Socialize the preliminary study results	30 min	Results presentation
Break	15	None
Activity 4: Apply SpeTion-SPrL	3 hours	Task document, SpeTion guide document, scenarios and result templates for approach execution
Activity 5: Post meeting	1 hour	Scope matrix, future projects document
Total time: 4:30 hours 15 min		

Table 4. 25. Case study activities summary

4.5.5 Execution

- Activity 1: This first part aims to socialize and contextualize in a general way, what the case study consists. To carry out this activity it is necessary to make an oral presentation in order to introduce and inform the case study participants, about the activities that will be developed, in addition to making known and clarifying some concepts used in it.
- Activity 2: This activity is guided by an oral presentation of SpeTion-SPrL so that participants have a method overview that will be applied later.
- Activity 3: This activity is guided by an oral presentation of the SpeTion-SPrL application results during the preliminary study which gives us the method application context and provides the input elements for its execution.
- Activity 4: As an initial part of the activity development, it is necessary to guide the participants in the method application and support material used. In the second part of this activity, the case study participants will apply SpeTion-SPrL using the guide to define the process line scope. As a result, participants must complete the artifacts defined in the approach.
- Activity 5: In this last activity, a meeting will be held with the case study participants to identify the following aspects:
    - Identify future projects in the same SPrL domain
    - The participants made a survey (See the survey in the Annex *N*, Section 1: Survey)

4.5.6 Analysis and Results

For the hypothesis validation, the data was taken from the results obtained from the templates that are the result of applying the SpeTion approach by employees (activity 4) and also data was taken from the survey results (activity 5). The survey responses were based on the Linkert scale, which is a form of measurement that allows to assess attitudes and know the conformity degree on a statement set. The measurement scale of the survey responses was defined as follows: value 1 for the totally disagree option, value 2 for the disagree option, value 3 for the neutral option (neither agree nor disagree), value 4 for the agreement option and value 5 for the option totally in agreement. Of the hypotheses initially drawn, the following null hypotheses were raised:

- $H.1_0$, $\boldsymbol{\pi}1 <= 60\%$, where $\boldsymbol{\pi}1$ is the perception percentage that evaluates the steps taken in order during the SpeTion-SPrL execution
- $H.2_0$, $\boldsymbol{\pi}2 <= 60\%$, where $\boldsymbol{\pi}2$ is the perception percentage that evaluates the elements that will really support the creation of future projects
- $H.3_0$, $\boldsymbol{\pi}3 <= 60\%$, where $\boldsymbol{\pi}3$ is the perception percentage that evaluates the inconsistencies identified in the resulting scope
- $H.4_0$, $\boldsymbol{\pi}4 <= 60\%$, where $\boldsymbol{\pi}4$ is the perception percentage that evaluates the gradually outputs materialized of the tasks for making up the scope

From the null hypotheses the following alternative hypotheses were obtained:

- H.1, $\boldsymbol{\pi}1 > 60\%$, where $\boldsymbol{\pi}1$ is the perception percentage that evaluates the steps percentage taken in order during the SpeTion-SPrL execution
- H.2, $\boldsymbol{\pi}2 > 60\%$, where $\boldsymbol{\pi}2$ is the perception percentage that evaluates the elements that will really support the creation of future projects
- H.3, $\boldsymbol{\pi}3 > 60\%$, where $\boldsymbol{\pi}3$ is the perception percentage that evaluates the inconsistencies identified in the resulting scope
- H.4, $\boldsymbol{\pi}4 > 60\%$, where $\boldsymbol{\pi}4$ is the perception percentage that evaluates the gradually outputs materialized of the tasks for making up the scope

To validate the hypotheses raised, the participants resolved the designed survey and filled out the templates, which were the result of applying the approach, the values obtained per metric are presented below.

For the systematicity see the Table 4.**26**:

Systematic			
	SS	**TS**	**Systematicity (%)**
Participant 1	0	21	100,00
Participant 2	0	21	100,00
Participant 3	3	21	85,71

Table 4.26. Systematicity values

Where, *SS* refers to the steps omitted by practitioners, this information was obtained from the steps that were not executed in the templates provided by the participants, and *TS*, to the Total steps of SpeTion in the scope analysis and design process.

For the useful see the Table 4.**27**:

Useful			
	SEWU	**SE**	**Utility (%)**
Participant 1	329	396	16,92
Participant 2	249	396	37,12
Participant 3	166	396	58,08

Table 4.27. Useful values

Where, *SEWU* refers to the scope elements that will be used in future projects, these values were taken from survey questions 1, 2 and 7, and *SE* refers to the total elements defined in scope, that is, context characteristics, features, suitability values (Scope matrix).

For the clarity see the Table 4. **28**:

Clear			
	ISE	**SE**	**Clarity (%)**
Participant 1	182	396	54,04
Participant 2	206	396	47,98
Participant 3	234	396	40,91

Table 4. 28. Clarity values

Where, *ISE* refers to the total elements of scope that are considered inconsistent, these values were taken from survey questions 4 and 6, and *SE* refers to the total elements defined in scope, that is, context characteristics, features, suitability values (Scope matrix).

For the tangibility see the Table 4.**29**:

Tangible			
	TO	**TA**	**Tangibility (%)**
Participant 1	5	5	0,00
Participant 2	5	5	0,00
Participant 3	5	5	0,00

Table 4.29. Tangibility values

Where, *TO* refers to the total generated output by the approach, these values were taken from templates filled out by participants due to the use of the approach and *TA* refers to the total task of the approach.

Of the values obtained in each indicator, we can say:

- For the systematicity indicator, the perception percentage for each participant is 100%, 100% and 85,71% respectively, that determined that the steps followed during the SpeTion-SPrL execution allow advanced in an orderly way according to its definition.
- For the utility indicator, the perception percentage for each participant is 16,92%, 37,12% and 58,08% respectively, that determined that the scope resulting from executing of SpeTion-SPrL supports the creation of future projects in the same domain, considering that the lower indicator values indicate better utility.
- For the clarity indicator, the perception percentage for each participant is 54,4%, 47,98% and 40,91% respectively, that determined that there are inconsistencies identified in the resulting scope, this, according to the analysis procedure set out above.
- For the tangibility indicator, the perception percentage for each participant is 0%, that determined that the scope, which is the result of executing SpeTion-SPrL is materialized through the outputs of the tasks.

According to the results of the indicators, it can be inferred for the specific hypotheses, that:

- H.1 can be accepted. In this way, it can be said that SpeTion-SPrL processes are properly conducted.
- H.2 can be accepted. In this way, it can be said that SpeTion-SPrL results are useful.
- With the results of the utility indicator with lower percentages than 60%, H.3. can be rejected. In this way, it can be said that the SpeTion-SPrL results aren´t clear.
- H.4. can be accepted. In this way, it can be said that SpeTion-SPrL artifacts are tangible.

With most specific hypotheses accepted (3 of 4) we can infer that the main hypothesis is accepted, determining that: The result of SpeTion-SPrl is suitable for determining the process line scope of GreenSQA (For the results obtained in the surveys, see Annex *N*, section 2: Results obtained).

4.5.7 Conclusions

This study confirms the SpeTion-SPrL approach is suitable for determining a process line scope, for this H1, H2, H3 and H4 were defined. Hypothesis H1 make reference to the user's perception that the steps followed during the SpeTion-SPrL execution allow advanced in an orderly way according to its definition, the hypothesis H2 make reference to the user's perception that determined that the scope resulting from executing of SpeTion-SPrL supports the creation of future projects in the same domain, the hypothesis H3 make reference to the user's perception that determined that there are inconsistencies identified in the resulting scope and the hypothesis H4 make reference to the user's perception that the scope, which is the result of executing SpeTion-SPrL is materialized through the outputs of the tasks. According to the validation of these four hypotheses, it can be said that the participants managed to perceive that the SpeTion-SPrL approach is not clear, since it has a conceptual complexity that does not allow an easy execution of the scope definition, complexity that remains evident since the academic experiment presented in Chapter 4, Section 4.3. However, it can be said that the SpeTion approach is systematic because it is properly conducted, useful because it helps in the definition of future projects and tangible because it is materialized through the outputs of the tasks.

4.5.8 Threads of validity

Construct validity: To minimize the subjectivity in the instruments supporting the collection of information from the case study, during its planning it is necessary that the instruments be submitted to validations with the participation of the researchers. Another identified threat of this type is the incorporation and management of new conceptual and language elements in the study development. In order to reduce this threat as part of the case study, an initial activity was planned in which socialized and contextualized to participants in the study details to achieve a unification of the concepts and language.

Internal validity: a validity threat of this type is the time invested for the study execution, being long sessions, participants in the final stages of the study can perceive tiredness which can influence the results. To try to mitigate this threat as part of the study, a break was defined in which there should be no communication between the participants. Another threat of this type is the participants selected by the company which was limited. The low number of participants that the company facilitates in the collection of information is internal validity threat. To try to control this, it was requested that the participants from the company be at least three project managers with good experience in testing processes.

External validity: due to in the case study development, there will be experts in processes by the group of researchers and the company participants, this is a threat because this type of staff is not available in all companies, in such a way that, they will have to supply this lack. In addition, to apply our approach it is necessary for companies to have mature processes and good document management in a way that allows identifying current and future products and projects, which will not be the case for many companies. The external validity of this study depends to some extent on the similarity of the context where the method will be applied with respect to the context of this case study.

Chapter 5

Conclusions y Future Work

This chapter concludes the project, unveil limitations and identify future work.

5.1 Conclusions

This thesis presents SpeTion-SPrL as a systematic and useful approach for scoping a Software Process Line in different contexts. The scoping method is based on the identification quantitative of the need for projects and products with respect to the process elements. It identifies the needs and their corresponding correlation with the process assets through a suitability level that allows quantitatively select and make adaptation decisions on the process elements. SpeTion-SPrL attending to the demands found since the literature review, it was built under the principles of systematic, integrality, properly conducted and materializable, whose support is determined by the different elements that make it up. It integrates elements of three main approaches: SCOPE, CASPER and SPL concepts, where it has been shown that its integration is feasible and practical, allowing to guide the scope definition and increase understanding and confidence in this concept and in its definition process. SpeTion-SPrl structures two processes that cover two moments related to the analysis and design, and scope use, so that these processes are coherent with the phases of the construction and use of the process lines, emphasizing that the scope definition is a holistic issue that should be considered in all process line engineering and not just in the construction phase.

SpeTion-SPrL was defined using systematic techniques and refined through its empirical evaluation. The first validation consisted in an empirical study to determine, in the context of the GreenSQA company. The empirical study was divided in two parts, the first one aimed was to know what are the advantages and disadvantages of applying SCOPE and CASPER in the practice (Part A), and the second one aimed to know the feasibility of an integrated approach for defining the scope (Part B), the results of which allowed us to said that the integration between SCOPE and CASPER were moderately feasible. The second validation consisted of testing the ease of use, usefulness, and reliability of the approach. For this, an experiment was developed with students from three Latin American universities. Hence, it was possible to verify that the approach lacks elements in its description that would allow a better understanding of its definition so that this allowed us to identify future improvements. In addition, it was perceived that there is a utility in the approach, through an organized and consistent approach. Likewise, due to the suitability results obtained, it can be considered a reliable and therefore systematic approach that can help determine the inclusion or exclusion of process features for a specific situation in a reasonable way. In the third evaluation, a case study was developed in the context of the GreenSQA company, with the purpose of evaluating its suitability from the process engineers team. From Case study results we can say that the SpeTion approach is systematic because it is properly conducted, useful because it helps in the definition of future projects and tangible because it is materialized through the outputs of the tasks.

The different results shown both in the experimentation and in the case study indicate that SpeTion-SPrL as an integrated approach adequately helps in defining the scope of Software Process Lines, it provided an approach that has processes, tasks, guidelines and examples which It addresses the two phases of Process Line Engineering, Domain Engineering, and Application Engineering, allowing not only that the materialization of the scope be reflected in the construction of the process lines, but also helps to make adaptation decisions when SPrL is in use. In addition, following the call of the academic community to increase the empirical evidence of the SPrL approach (Carvalho & Chagas, 2014), (Chen, Babar, & Ali, 2009), (Blum, Simmonds, & Bastarrica, 2015), (Schramm, Dohrmann, & Kuhrmann, 2015) with the development of chapter 4, evidence is provided in the application of this approach, specifically in the scope definition, in addition, evidence is provided in the comparison of two SPrL scoping approaches that highlight the advantages and disadvantages of each one of them. However, the evidence reported in this thesis is specifically of the definition of the scope, therefore, it is not sufficient to provide complete evidence in the whole approach, but it is a support to gradually achieve maturity in this area, which allows the industry benefits from the application of the approach. Finally, there is a technological gap in this thesis, SpeTion-SPrL is a proposal that includes templates, tables, and calculations; elements that are not included in integrated technological support, therefore, to validate this thesis, we have used a minimum infrastructure as support tools.

5.2 Limitations

- Although SpeTion-SPrL tries to serve as an approach to define the scope in lines of software processes for the software industry, it uses a minimum infrastructure of support tools.
- The scope is represented in a simple spreadsheet where there is a set of cross-references between the elements of the context and the process
- Although the scope approach presented in this thesis materializes the scope through the definition of a matrix, it no is easy to
- SpeTion-SPrL scope definition is linked only to the domain of software processes leaving other domains where SpeTion-SPrL can be of help.

5.3 Further work

Several further work to be considered as follow:

- SpeTion-SPrL still requires more empirical evidence. It is part of the dynamics in software engineering that empirical evidence requires the efforts of many organizations and many years to become sufficient to achieve models assimilated by the software industry.
- The approach needs to incorporate technological aspects that facilitate in an integrated way the materialization of the scope and its application in the industry. Also, it needs to incorporate an automatic data analysis in such a way that it allows to better support the decision-making process adaptation.
- To incorporate a means of scope visualization that allows to easily understand its composition and behavior.
- A scope model needs to propose that considers all the elements essential to support its conceptual definition and thus be more easily analyzed, understood and evolved.
- Further work is about SPL scoping using the process family supported by the derivable product. So, the process family is another level for defining the scope, so SPrL Scope helps to the salesmen, consultants, and engineers to evaluate areas inside the 4.0 Industry where the SPL could derive attractive products

Bibliography

Acuña, S. T., & Ferré, X. (2001). *World Multiconference on Systemics, Cybernetics and Informatics, ISAS-SCIs 2001, Orlando, Florida, USA , I: Information Systems Development*.

Ahn, Y., Ahn, H., & Park, S. (2003). Knowledge and case-based reasoning for customization of software processes - a hybrid approach . *International Journal of Software Engineering and Knowledge Engineering*, 293-312.

Alcântara, P., Brittoa, R., Andrade, R., Almeida, J., & Ayala, W. (2016). A hybrid approach to suggest software product line portfolios. *Applied Soft Computing Journal, vol. 49*, 1243–1255.

Aleixo, F. A., Kulesza, U., & Oliveira Junior, E. A. (2013). Modeling Variabilities from Software Process Lines with Compositional and Annotative Techniques : A Quantitative Study. In *Product-Focused Software Process Improvement* (pp. 153-168). Springer Berlin Heidelberg.

Alsawalqah, H., Kang, S., & Lee, J. (2014). A method to optimize the scope of a software product platform based on end-user features. *The Journal of Systems & Software*, 79–106.

Armbrust, O., Ebel, J., Hammerschal, U., Münch, J., & Thoma, D. (2008). Experiences and Results from Tailoring and Deploying a Large Process Standard in a Company. *Software Process: Improvement and Practice SPIP*, 301-309.

Armbrust, O., Katahira, , M., Yuko , M., Munch, J., Nakao, H., & Ocampo, A. (2009). Scoping Software Process Lines. *Journal Software Process: Improvement and Practice - Examining Process Design and Change*, 181-197 .

Armbrust, O., Katahira, M., Yuko, M., Jürgen, M., Haruka, N., & Ocampo, A. (2008). Scoping Software Process Models - Initial Concepts and Experience from Defining Space Standards. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 395-406.

Balbino, M., Santana de Almeida, E., & Meria, S. (2011). An agile scoping process for Software Product Lines. *In SEKE 2011 - Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering*, 717–722.

Barreto, A. S., Murta, L., & Cavalcanti da Rocha, A. R. (2011). Software Process Definition : a Reuse-based Approach. *Journal of the Universal Computer Science*, 1765-1799.

Bartholdt, J., & Becker, D. (2012). Scope Extension of an Existing Product Line. *In Proceedings of the 16th International Software Product Line Conference, Volume 1*, 275–282.

Bastarrica, M. C., Simmonds, J., & Silvestre, L. (2014). *A Megamodel for Process Tailoring and Evolution.*

Baud, J., & Schmid, K. (1999). A systematic approach to derive the scope of software product lines. *International Conference on Software Engineering (IEEE Cat. No.99CB37002)*, 34-43.

Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change.* Addison-Wesley Professional, 2nd edition. 12, 21.

Blum, F. R., Simmonds, J., & Bastarrica, M. C. (2015). Software process line discovery. *Conference Software and System Process*, pp. 127–136.

Bosch, J. (2003). Maturing architectures and components in software product lines. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),vol. 2693*, 246–258.

Brereton, P., Kitchenham, B., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw., vol. 80,*, 571–583.

Bunge, M. (2002). Philosophy of Science. *México: From Problem to Theory. New Brunswick, Transaction. Siglo Veintiuno Editores*.

Bustard, D., & Keenan, F. (2005). Strategies for Systems Analysis: Groundwork for Process Tailoring. *Engineering of Computer-Based Systems ECBS, 12th IEEE International Conference and Workshops*.

Camacho, M., Alvarez, F., Hurtado, J., & Ruiz, P. (2019). Importance of Teamwork in Scoping of Software. *IEEE*.

Carvalho, D. D., & Chagas, L. F. (2014). Software process lines: A systematic literature review. *Software Process Improvement and Capability Determination, Vilnius*, pages 118–130.

Cass, A., Lerner, A., & McCall, E. (2000). Little JIL/Juliette: A Process Definition Language and Interpreter. . *Proceedings of the 2000 International Conference on Software Engineering*, 754 - 757.

Cavalcanti, Y. C., Machado, I., Mota, P. A., Lopes, L., Santana, E., & Lemos, S. (2011). Towards Metamodel Support for Variability and Traceability in Software Product Lines. *In Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems*, 49–57.

Chen, L., Babar, M., & Ali, N. (2009). Variability management in software product lines: A systematic review. In Intl. Software Product Line Conference, pages 81–90. Carnegie Mellon University, 2009. *In Intl. Software Product Line Conference*, 81-90.

Clements, P. (2002). On the Importance of Product Line Scope. *Software {Product-Family} Engineering. Springer-Verlag Berlin Heidelberg*, 102-113.

Clements, P., & Northrop, L. (2001). *Software product lines: practices and patterns.* Addison-Wesley.

Cockburn, A. (2002). Agile Software Development. *Adisson-Wesley*.

Costa, D., Nogueira Teixeira, E., & Lima Werner, C. (2018). Software Process Definition using Process Lines: A Systematic Literature Review. *XLIV Conferência Latino-americana de Informática*.

Cruz, J., Santos, P., Britto, R., Rabelo, R., Ayala, W., Soares, T., & Mota, M. (2013). Toward a hybrid approach to generate Software Product Line portfolios. *In 2013 IEEE Congress on Evolutionary Computation, CEC 2013*, 2229–2236.

Cvetković, R., & Nešković, S. (2010). An approach to defining scope in software product lines for the telecommunication domain. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 555–558.

Dias De Carvalho, D., Chagas, L. F., & Reis, C. A. (2014). Definition of software process lines for integration of Scrum and CMMI. *CLEI 2014*.

Dongwon, K., In-Gwon, S., Seunghun, P., Doo-Hwan, B., Hoon-Kyu, K., & Nobok, L. (2008). A Case Retrieval Method for Knowledge-Based Software Process Tailoring Using Structural Similarity. *Software Engineering Conference, 15th Asia-Pacific*, 51-58.

Fei, D., & Tong, L. (2007). Tailoring Software Evolution Process. *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*.

Feiler, P. H., & Humphrey, W. S. (1993). Software process development and enactment: concepts and definitions. *Proceedings of the Second International Conference on the Software Process-Continuous Software Process Improvement*, 28-40.

Fernandes, L. d., Lemos, S., & Santana Almeida, E. (2012). CodeScoping : A Source Code Based Tool to Software Product Lines Scoping. *Proceedings - 38th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA*, 101-104.

France, R., & Rumpe, B. (2007). Model-driven Development of Complex Software: A Research Roadmap. *FOSE '07 2007 Future of Software Engineering*, 37-54.

Funk, P. (2000). Reuse, Validation and Verification of System Development Processes. .

Gillain, J., Faulkner, S., Heymans, P., Jureta, I., & Snoeck, M. (2012). Product Portfolio Scope Optimization Based on Features and Goals. *Proceedings of the 16th International Software Product Line Conference - Volume 1. ACM, New York, NY, USA*, 161–170.

Ginsberg, M., & Quinn, L. (1994). Process Tailoring and the software Capability Maturity Model. .

Golpayegani, F., Azadbakht, K., & Ramsin, R. (2013). Towards process lines for agent-oriented requirements engineering. *IEEE EuroCon 2013*, 550-557.

Hanssen, G., Westerheim , H., & Bjørnson, F. (2005). Tailoring RUP to a defined project type: A case study. *PROFES'05 Proceedings of the 6th international conference on Product Focused Software Process Improvement*, pp. 314-327.

Henninger, S., & Baumgarten, K. (2001). A case-based approach to tailoring software processes. *Proceedings of the 4th International Conference on Case-Based Reasoning . Lecture Notes in Computer Science*, 249-262.

Heradio-Gil, R., Fernandez-Amoros, D., Cerrada, J. A., & Cerrada, C. (2011). Supporting commonality-based analysis of software product lines. *IET Software., vol. 5, No. 6*, 496–509.

Heradio-Gil, R., Fernandez-Amoros, D., Cerrada, J., & Cerrada, C. (2011). Supporting commonality-based analysis of software product lines. *IET Software*, 496-509.

Hollenbach, C., & Frakes, W. (1996). Software Proeess Reuse in an Industrial Setting. *Proceedings of Fourth IEEE International Conference on Software Reuse. Orlando, FL, USA*.

Hossein, S., & Natsu, C. (1997). Characterizing a Software Process Maturity Model for Small Organizations. *ACM SIGICE Bulletin, 23*, 2-11.

Humphrey, W. S. (1989). Managing the Software Process. *Addison-Wesley Longman*.

Hurtado, J. A., & Bastarrica, M. C. (2012, junio). Building software process lines with CASPER. *IEEE*, 170-179. doi:10.1109/ICSSP.2012.6225962

Hurtado, J. A., & Bastarrica, M. C. (2012). Building Software Process Lines with CASPER. *International Conference on Software and Systems Processes. ICSSP*.

Hurtado, J. A., Bastarrica, M. C., Ochoa, S. F., & Simmonds, J. (2013). MDE software process lines in small companies. *Journal of Systems and Software, 86*, 1153-1171.

Hurtado, J., & Bastarrica, C. (2009). Process Model Tailoring as a Means for Process Knowledge Reuse. *In Proceedings of the 2nd Workshop on Knowledge Reuse. Falls Church Virginia, USA*.

ISO. (2008). ISO/IEC 12207 Information Technology / Software Life Cycle Processes. *International Organization for Standardization*.

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). Unified Software Development Process.

John, I. (2010). Using Documentation for Product Line Scoping. *IEEE Software, vol. 27, No. 3*, 42–47.

John, I., & Eisenbarth, M. (2009). A Decade of Scoping: A Survey. *Proceedings of the 13th International Software Product Line Conference*, (pp. 31-40). Pittsburgh, PA, USA.

John, I., Knodel, J., Lehner, T., & Muthig, D. (2006). A Practical Guide to Product Line Scoping. *10th International Software Product Line Conference (SPLC'06)*, 3-12.

Kishi, T., Noda, N., & Katayama, T. (2002). A Method for Product Line Scoping Based on a Decision-Making Framework. *International Conference on Software Product Lines* , 348-365.

Klaus, P., Günter, B., Linden, V., & Frank, J. (2005). *Software product line engineering: foundations, principles and techniques.* Springer.

Kuhrmann, M., Ternité, T., Friedrich, J., Rausch, A., & Broy, M. (2016). Flexible software process lines in practice: A metamodel-based approach to effectively construct and manage families of software process models. *Journal of Systems and Software, Volume 121*, 49-71.

Kuloor, C., & Eberlein, A. (2003). Aspect-oriented requirements engineering for software product lines. *10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*, 98-107.

Lanzen, A., Fontana, R., Paludo, M., Malucelli, A., & Reinehr, S. (2015). Scoping automation in software product lines. *In 17th International Conference on Enterprise Information Systems, (ICEIS 2015), vol. 2*, 82–91.

Lee, .., & Lee, D. H. (2012). A market-driven product line scoping. *Studies in Computational Intelligence, vol. 377*, 27-46.

Lee, J., & Lee, N. (2006). Least modification principle for case-based reasoning: a software project planning experience. *Expert Systems with Applications, Elsevier*, 190–202.

Lee, J., Kang, S., & Lee, D. (2010). A COMPARISON OF SOFTWARE PRODUCT LINE SCOPING APPROACHES. *International Journal of Software Engineering and Knowledge Engineering. Vol. 20, No. 5*, 637–663.

Lobsitz, R. (1996). A Method for Assembling a Project-Specific Software Process Definition. *Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences (HICSS'96)*, 722-730.

Magdaleno, A., Araujo, R., Werner, C., & Alves Batista, F. (2015). A Practical Experience of a Software Process Line Creation. *18th Conferencia Iberoamericana en Software Engineering. CIbSE*, 95-108.

Moraes, M. D., Santana de Almeida, E., & Romero de Lemos Meira, S. (2009). A Systematic Review on Software Product Lines Scoping. *6th Experimental Software Engineering Latin American Workshop Organization*, 63-72.

Nöbauer, M., Seyff, N., & Groher, I. (2014). Similarity Analysis within Product Line Scoping: An Evaluation of a Semi-automatic Approach. *Advanced Information Systems Engineering. CAiSE 2014. Lecture Notes in Computer Science, vol 8484. Springer, Cham* , 165-179.

Nöbauer, M., Seyff, N., Groher, I., & Dhungana, D. (2012). A Lightweight Approach for Product Line Scoping. *In 2012 38th Euromicro Conference on Software Engineering and Advanced Applications*, 105–108.

Noor, M. A., Grünbacher, P., & Briggs, R. O. (2007). A collaborative approach for product line scoping: A case study in collaboration engineering. *In Proceedings of the IASTED International Conference on Software Engineering*, 216–223.

Northrop, L., Clements, P., Bachmann, F., Bergey, J., Chastek, G., Cohen, S., . . . O'Brien, L. (2007). A Framework for Software Product Line Practice, Version 5.0,. *SOFTWARE ENGINEERING INSTITUTE* | *CARNEGIE MELLON UNIVERSITY* .

Ocampo, A., Bella, F., & Münch, J. (2005). Software Process Commonality Analysis. *In Software Process: Improvement and Practice. Special issue on ProSim 2004, The 5th International Workshop on Software Process Simulation and Modeling, Edinburgh, Scotland, 10*, 273-285.

Oliveira, E., Pazin, M., Gimenes, M., & Kulesza, , U. (2013). SMartySPEM : A SPEM-Based Approach for Variability Management in software process lines. *14th International Conference, PROFES 2013, Paphos, Cyprus, Proceedings*, 69-183.

Osterweil, L. (1987). Software processes are software too. *In ICSE '87: Proceedings of the 9th International Conference on Software Engineering, IEEE Computer Society Press, Los Alamitos, CA, USA*, 2-13.

Park, S. Y., & S. D. Kim. (2005). A systematic method for scoping core assets in product line engineering. *12th Asia-Pacific Software Engineering Conference (APSEC'05)*, 8-17.

Park, S., Naa, H., Parka, S., & Sugumaranb, V. (2006). A semi-automated filtering technique for software process tailoring using neural network. . *Expert Systems with Applications*, 179-189.

Pedreira, O., Piattini, M., Luaces, R., & Brisaboa, N. (2007). A Systematic Review of Software Process Tailoring. *Software Engineering Notes SIGSOFT*, 1-6.

Ralyté, J., Deneckère, R., & Roll, C. (2003). Towards a generic model for situational method engineering . *CAiSE 2003, LNCS 2681, Springer-Verlag*, 95-110.

Ramírez, Y. L., Lasso, S. P., Ruiz, P. H., & Hurtado, J. A. (2018). Customizable Software Processes: An Exploratory Case Study in a Small Organization. *13 Conferencia Colombiana de Computación*.

Rombach, D. (2005). Integrated Software Process and Product Lines. *ISSN 0302-9743 Volume 3840/2006. book Unifying the Software Process Spectrum. ISBN 978-3-540-31112-6.*, 83-90.

Rossel, P., Herskovic, V., & Ormeño, E. (2016). Creating a family of collaborative applications for emergency management in the firefighting sub-domain. *Information Systems Frontiers, vol. 18, no. 1,* , 69–84.

Rouille, E., Combemale, B., Barais, O., Touzet, D., & Jezequel, J. (2012). Leveraging CVL to manage variability in software process lines. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, 148-157.

Ruiz, P. H., & Hurtado, J. A. (2013). A Software Process Line Based on the Unified Process. *Computing Congress (CCC) 978-1-4673-1475-6 , 2012 7th Colombian*.

Ruiz, P. H., & Hurtado, J. A. (2016). Framework Based on Software Process Lines for Tailoring of Software Process Models. *Conferencia Colombiana de Computación*, 2.

Ruiz, P. H., & Hurtado, J. A. (2017). Scope Determination in Software Process Lines. *CCC*.

Ruiz, P. H., Camacho, C., & Hurtado, J. A. (2018). A canonical software process family based on the Unified Process. *Revista Scientia et Technica , 23, septiembre de 2018. ISSN 0122-1701*(03).

Ruiz, P. H., Camacho, M. C., & Hurtado, J. A. (2018). A Comparative Study for Scoping a Software Process. *2018 ICAI Workshops (ICAIW)*.

Ruiz, P. H., Hurtado, J. A., & Camacho, M. C. (2014). Analizando la Variabilidad Del Proceso Unificado con la Tècnica de Washizaki. *Gestión de la Técnologia Informática*.

Ruiz, P., & Hurtado, J. (2012). A software process line based on the Unified Process. *2012 7th Colombian Computing Congress (CCC)*, 1-6.

Ruiz, P., & Hurtado, J. (2019). An Empirical Study for scoping a Software Process Line for Testing Services.

Ruiz, P., Agredo, V., Mon, A., & Hurtado, J. (2019). An experiment in scoping definition in software process lines. *Ingeniare. Revista chilena de ingeniería*.

Ruiz, P., Camacho, C., Marques, M., Hurtado, J., & Bastarrica, M. (2019). Software Products Lines as Support in Software Process Lines scope definition.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *(D. Sjoberg, Ed.) Empirical Software Engineering,* , 131-164.

Santos, R., Oliveira, T., & Abreu, F. (2015). Mining software development process variations. *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 1657-1660.

Schmid, K. (2000). Scoping software product lines. *Proceedings of the First Software Product Lines.* (pp. 513-532). New York: Springer Science+Business Media .

Schmid, K. (2000). Scoping software product lines. *Proceedings of the First Software Product Lines.* (pp. 513-532). New York: Springer Science+Business Media .

Schmid, K. (2002). A Comprehensive Product Line Scoping Approach and Its Validation. *Proceedings of the 24th International Conference on Software Engineering*, 593-603.

Schmid, K., & Gacek, C. (2000). Implementation Issues in Product Line Scoping. *Software Reuse: Advances in Software Reusability*, 170-189.

Schmid, K., & Schank, M. (2000). Pulse-beat — a decision support tool for scoping product lines. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 65–75.

Schmid, K., & Verlage, M. (2002). The economic impact of product line adoption and evolution. *IEEE Softw., vol. 19, no. 4.* , 50–57 .

Schramm, J., Dohrmann, P., & Kuhrmann, M. (2015). Development of flexible software process lines with variability operations: a longitudinal case study. *in Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, P. 13.

Schwaber, K. (1997). Scrum Development Process. *in OOPSLA Business Object Design and Implementation Workshop, J. Sutherland, et al, Springer: London.Scott, L., Jeffery, R., Carvalho, L., D'Ambra, J.*

SEI. (2006). CMMI for Development. *Software Engineering Institute, Tech.Rep. CMU/SEI-2006-TR-008*.

Sellers, B., & Ralyté, J. (2010). Situational Method Engineering: State-of-the-Art Review. . *Journal of Universal Computer Science, vol. 16, no. 3*, 424-478.

Simidchieva, B., Clarke, L., & Osterweil, L. (2007). Representing Process Variation with a Process Family. *In qing Wang, Dietmar Pfahl, and David M. Raffo, editors, ICSP´2007, volumen 4470 of LNCS*, 109-120.

Simmonds, J., Bastarrica, M., Silvestre, L., & Quispe, A. (2013). Variability in software process models: Requirements for adoption in industrial settings. *2013 4th International Workshop on Product LinE Approaches in Software Engineering, PLEASE 2013 - Proceedings*, 33-36.

Standish Group. (2012). Chaos, Standish Group: West Yarmouth.

Teixeira, E., De Mello, R., Motta, R., Werner, C., & Vasconcelos, A. (2015). Verification of software process line models: A checklist-based inspection approach. *CIBSE 2015 - XVIII Ibero-American Conference on Software Engineering*, 81-94.

Ternité, T. (2009). Process lines: A product line approach designed for process model development. *Conference Proceedings of the EUROMICRO*, 173-180.

Villela, K., Dörr, J., & John, I. (2010). Evaluation of a Method for Proactively Managing the Evolving Scope of a Software Product Line. *Requirements Engineering: Foundation for Software Quality. REFSQ 2010. Lecture Notes in Computer Science, vol 6182. Springer, Berlin, Heidelberg*, 113-127.

Washizaki, H. (2006). Building Software Process Line Architectures from Bottom Up. *Product-Focused Software Process Improvement*, 415-421.

Wnuk, K., Regnell, B., & Karlsson, L. (2009). What happened to our features? Visualization and understanding of scope change dynamics in a large-scale industrial setting. *Proceedings of the 17t h IEEE International Conference on Requirements Engineering*, 89-98.

Xu, P. (2005). Knowledge Support in Software Process Tailoring. *Management Proceedings of the 38th Hawaii International Conference on System Sciences, 03-06 Jan.*, 87.

Yin, R. K. (1994). Discovering the future of the case study method in evaluation research. *1994 Published by Elsevier Inc.*, 283-290.

Yoon, I., Min, S., & Bae, D. (2001). Tailoring and Verifying software Process. *Eighth Asia-Pacific Software Engineering Conference (APSEC'01).*, 202.