A canonical software process family based on the Unified Process

Una familia de procesos software canónica basada en el Proceso Unificado

Pablo H. Ruiz[1], Vanessa Agredo-Delgado[1], Marta Cecilia Camacho[2], Julio A. Hurtado[3]
[1]*Programa de ingeniería de Sistemas grupo de investigación TIC Unicomfacauca, Corporación Universitaria Comfacauca, Popayán, Colombia*
pruiz@unicomfacauca.edu.co,vagredo@unicomfacauca.edu.co
[2]*Programa de ingeniería Informática grupo de investigación y desarrollo en Informática, Institución Universitaria Colegio Mayor del Cauca, Popayán, Colombia*
ccamacho@unimayor.edu.co
[3]*Programa de ingeniería de Sistemas grupo de investigación [a] IDIS, Universidad del Cauca, Popayán, Colombia*
ahurtado@unicauca.edu.co

***Abstract*— The Unified Process (UP) is a processes framework widely known and used by the software industry and academic community. UP was developed under the conception of a universal application, but even a single process is not appropriate to address all development projects since optimal process depends on the particularities of each project or organization. Frequently, many process components need to be adapted to the organization or project needs, a complex task because the experience and knowledge required. Perform adaptation in an inappropriate way is prone to error due to the difficulties for taking tailoring decisions. The paper goal is proposing a canonical software processes family based on the Unified Process to support software process tailoring. Our methodology was based on the meta-process CASPER to build a software process family and we use the Case Study method as a strategy for validating the canonical process family empirically. As a result, we show a canonical software processes family, which will serve as a platform for determining a process family based on Unified process in a small software organization. This canonical family has the advantage that uses a tailoring mechanism following a transformation strategy that encapsulates tailoring decisions in order to systematize this activity. In this work, we show the initial results in the applicability of a canonical software process family as base in the building the software process families based on UP in the small companies context, and also we conclude that the UP is a process framework with spread spectrum, which limited its formulation like a general software process family, due to that a process family requires determining carefully the application domain known as software process family scope.**

Key Word — Software process family, software process, software process tailoring, Unified Process.

I. INTRODUCTION

In an attempt to apply the software processes, the companies usually do not fit the process standard to their own context or a specific project requirement; this is mainly due to the complexity and required knowledge for adaptation. Software industry and software engineering research have given continuously a good number of successful software processes models. However, it is improbable that each of these proposals, as such, has been equally useful for any organization or project, considering that organizations are different, and projects within the same organization are unique [1]. A lot of process components need to be adapted to specific contexts, both organization and project [2]. The Unified Process (UP) is a software processes framework that has been widely used in the academic and the software industry [3], [4], but it is a generic framework that tries to be applied to multiples contexts of software projects and organizations which must be adapted to meet the diverse characteristics of both companies and projects [5], [1]. Adapting UP requires a careful and complex work [4], achieve its objectives is one of its main challenges because UP does not address clearly its adaptation to specific needs [6], [3]. Due to the adaptations that enterprises require, the UP tailoring is not a cost-effective solution, it subtracts practical applicability to the process, adding risks and additional costs by using an unsuitable process. Several UP adaptations are not repeatable and therefore each new adaptation is different, thus it requires experience, is error prone and requires a huge effort. Ones of the effective techniques that provide better benefits and expectations in software process tailoring are the Software Processes Lines and Software Process Families [7] [8], due to they allow to reduce the effort in adapted processes generation through planned reuse of process assets [7], [8].

Fecha de Recepción: 03 de diciembre de 2017
Fecha de Aceptación: 22 de agosto de 2018

This paper shows, a canonical Software Processes Family based on Unified Process (SoftProF-UP) which one can become a basis for software companies to define their own processes family and these can take advantage of a systematic adaptation mechanism. The remainder of this paper is organized as follows: Section II presents the research work related. Section III presents the methodology for building canonical SoftProF-UP. Section IV presents the main elements our proposal. Section VI illustrative case study determine the applicability of the canonical family. Section VII provides conclusions and future work.

II. RELATED WORK

A. Unified Process Tailoring Approaches

The UP in its environment discipline tries to provide a tailoring guide [5], where the process configuration is described; however, it does not clearly define how the process should be configured according to the project's characteristics, it limited only to provide general information that may be useful, but it is not sufficient for tailoring.

The authors [9] present a meta-model to describe a set of elements and relationships necessary to ensure consistency in tailoring, including new elements and relationships. In addition, they present a prototype of tool ProTTo (Process Tailoring Tool) to support the tailoring process. The work focus is the tailoring mechanisms definition, but not how the approach taking into account information from the organizations or projects context.

A generic model for the tailoring software processes is presented by [10], where the tailoring is made based on the project´s characteristics and doing manually. It is a model that was evaluated by a company where the RUP planning and management discipline was adapted. The authors [11] propose a framework to assist tailoring process, where knowledge and experience are organized in an appropriate way that will benefit the future projects tailoring, in addition, it shows a case study in a company in which RUP has been adopted as an organizational process. Ruiz et al. [12] present a tailoring guide to adapt the UP to projects specific for small organizations. However, it is an approach that, despite using SPEM 2.0 (Software Process Engineering Meta Model) [13] and EPFC[1], the process is not automatically tailored

B. Unified Process Adaptations and extensions

The UP adaptations and extensions allow evidence the importance that the software engineering community has given to the adequate processes definition to cover different contexts. Among the most relevant are:

- Agile UP (AUP), a simplified version of Rational Unified Process (RUP), it describes a simple and easy-to-understand approach to software development using RUP concepts and techniques. AUP adopts many of the XP agile techniques and other agile processes but retains some of the formality of RUP [14].
- Basic Unified Process (BUP): It is a RUP simplified version perfected for small projects. BUP retains the essential RUP characteristics. In this version, most optional parts of RUP have been excluded and many elements have been merged. The result is a much simpler process that remains true to the RUP principles which are aimed at projects and small teams [15].
- Open Unified Process (OpenUP) and Open Unified Process Basic (OpenUP / Basic): It is a minimum and sufficient light version of the UP, it does not provide guidelines for all the elements that are handled in a project, but it does have the basic components that can be the basis for specific processes. OpenUP / Basic is the agilest and lightweight form of OpenUP that has its roots in an open source donation of the BUP process contents [16].
- Enterprise Rational Unified Processes (EUP): It is a RUP extension. The support lack for the system and the eventual withdrawal of a software system by RUP are some of the elements that EUP tries to cover. To address this, EUP adds two phases and seven new disciplines [17].

C. Families and Process Lines

According to Simidchieva et al. [18] a processes family is the different agreed variations set 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 large processes adaptation through the exchange of components depending on the circumstances of execution. Cass et al. [19] presented a way to define and represent processes families with Little-Jil applying the techniques of software products families to direct the processes variation.

Rombach et al. [20] propose 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 products set, process requirements and project constraints.

D. Process Lines using MDE techniques

[1] EPFC – Eclipse Process Framework Composer

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 [21]. 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 particular processes taking into account the organization and/or project context. 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 instantiation strategies in software process tailoring. Hurtado et al. [22] supports adaptation at the conceptual level and provides mechanisms to make a 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 transformation model.

III. METHODOLOGY

To CASPER -Context Adaptable Software Process EngineeRing [23] as a methodological framework, view Figure 1. In CASPER is defined several activities in domain engineering and application engineering which ones allowed us to guide the building of the canonical SoftProF-UP. However, other additional methods were used in this definition, that will show later. To capture the specific software projects characteristics, CASPER suggests defining a Context Model (CM), for our case, it required search several UP application contexts. It was performed as a systematic review [24] and the Context Definition Method - PCDM [25]. The PCDM use as input, information gathered by the systematic review execution and as output, the family CM as an instance of SPCM. In order to analyze the SoftProF-UP variability, we follow the Washizaki strategy [8] comparing and determining the models set variability based on the UP process gathered from the literature. As result, a process architecture was defined for supporting the processes family incorporating common and variables components. This result was reported in [26]. The tailoring activity is defined at two stages. First, a transformation strategy based on MDE [22] is implemented as part of the domain engineering process, with the aim of systematizing the software process models creation that matches specific project's needs. And second, execution of tailoring strategy where project leader defines a project-specific context and executes the transformation strategy, generating a context-adapted process.

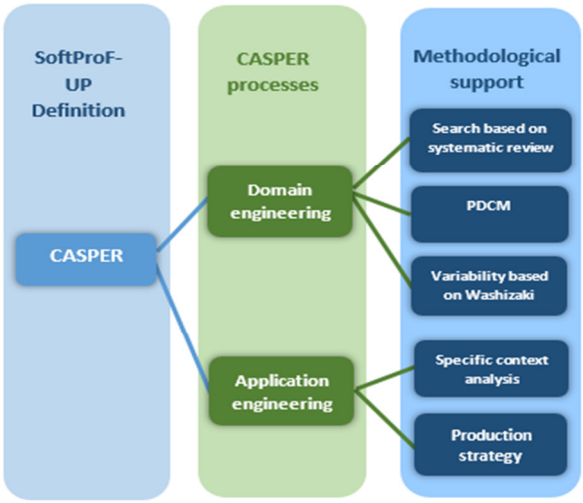

Figure 1. Methodology approach for SoftProF-UP definition

IV. SOFTPROF- UP: CANONICAL MODEL

A. Conceptual Structure

This work proposes a canonical software process family (SoftProF-UP) and a systematic mechanism for tailoring the Unified Process to particular contexts providing basic processes set adequate to project contexts. Its main objective is to help and serve to process engineers as a starting point for defining a software processes family based on UP that fit specific contexts. It uses three main approaches: process modeling and its variability, context modeling and a model transformation as an adjustment mechanism. The family is presented in Figure 2.

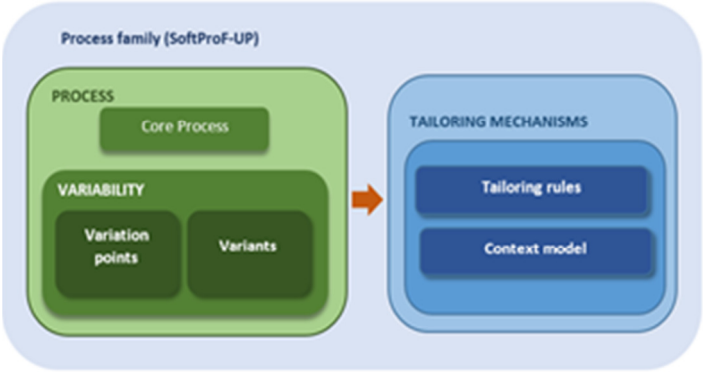

Figure 2. SoftProF-UP structure

- *Core Process* defines the process elements that are common as part (mandatory basis) in all possible configurations.
- *Process Variability* allows providing a process family with flexibility, adaptability, and reuse [8]. The variability in SoftProF-UP is defined by variation points and variants.
- Software process family includes a process reference model (core and variability).

- *Context Model* allows defining project context characteristics to determine the most appropriate software process model to support software development.
- *Tailoring rules* allow relating attributes of the canonical context model with the process variability. The tailoring rules are responsible for making the adaptation of a reference process model to an adapted process model taking into account the specific characteristics embodied in the context model configuration.
- *Tailoring mechanism* brings together, in an orderly manner, the elements that participate in the adaptation strategy, software reference process model (core and variability), context model and adaptation rules; inputs to define the software process according to specific needs. The adaptation is materialized using model driven engineering techniques - MDE.

Below is shown the different SoftProF- UP elements.

B. Software process family of the implementation discipline

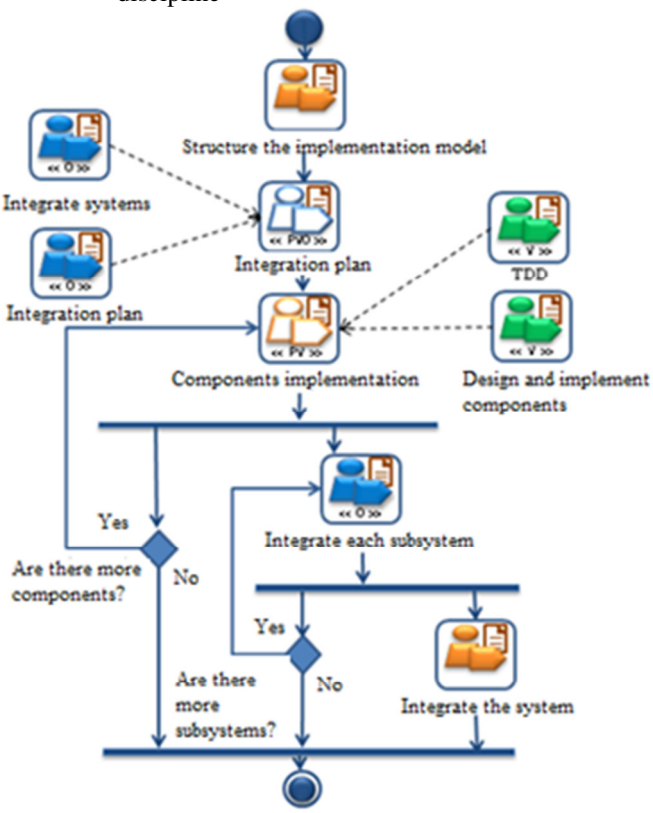

Figure 3. SoftProF-UP implementation discipline workflow.

The SoftProF-UP implementation discipline workflow is shown in Figure 3. Where the activity "Structure the Implementation Model" is part of the core process and has the purpose of structuring the implementation model by identifying subsystems set that can be developed with relative independence. The "Integration Plan" is optional and has two optional variants described by "Integrate Systems" and "Integration Plan". The purpose of the "Integrate Systems" activity is to create a plan for the building's integration describing the necessary constructions and the requirements of each construction, as well as integrating each construction. The "Integration Plan" activity has the purpose of planning the integration of the system focusing on the subsystems that must be implemented and the order in which they should be integrated into the current iteration. The activity "Components Implementation ", is a variation point that has two variants activities "TDD" and "Design and Implements Components". The activity "TDD" has the purpose of developing software subsystems but first writing the tests and refactoring or improving through unit tests. The "Design and Implement Components" activity has the purpose of implementing the requirements and ensuring that the subsystems comply with their functionality, that is, that the requirements are correctly implemented by the components or subsystems. The following activity "Integrate each Subsystem" is optional and its purpose is to integrate the changes from multiple implementations to create a new version of an Application Subsystem. The "Integrate the System" activity has the purpose of integrating the application subsystems to create a new compatible and complete system version.

The Canonical Context model (CCM) purpose is to identify the particularities of the project or organization that may affect the process tailoring. The dimensions grouping context attributes and attributes values that have the purpose of providing determined values set that can be assigned to the context attributes at the time of making a CCM configuration, and of this way to respond to the needs and particular project characteristics or an organization.

The dimensions considered in the CCM are: a) Project dimension, which aims to project characterize taking into account context attributes that pretend to condition the project characteristics with attributes such as: Business context, Project type, Project priority and Restrictions. b) Development Dimension, they have as objective to characterize the project taking into account aspects that condition the project rigor such as: Type of application, Criticality, Project Size. c) Team Dimension aims to gather context attributes that can determine the team characteristics responsible for the project development such as: Size and geographical distribution of the team. e) Team experience, with this dimension it is intended to provide elements to characterize the project with respect to the experience that the team has, with respect to the experience in the Process, experience in the Application Domain and Technical Experience. The Canonical Context Model with its respective dimensions, attributes and values are shown in Figure 4.

C. Canonical Context Model

Project dimensión	**Values**
Business context	{Work contract, speculative, commercial, internal projects}
Project Type	{Evolution, speculative, internal, pre studies, maintenance, projects with business modeling}
Project priority	{Quality90 Time10, Quality70 Time30, Quality50 Time50, Quality30 Time70, Quality10 Time90}
Restrictions	{Approved standards, contractual requirements, available tools, schedule, budget}
Development dimension	
Application type	{Mainstream, push-button, greenfield projects, critical mission}
Criticity	{Comfort loss, low losses, moderate damage, big losses, life risk}
Project size	{Small, Medium, Large}
Team dimension	
Team size	{Small, Medium, Large}
Geographical distribution	{Same office, same building different offices, same city and company different buildings, same city different company, different cities}
Dimension of team experience	
Experience in the process	{ None, minimal, good, excellent}
Experience in the application domain	{ None, minimal, good, excellent}
Technical experience	{ None, minimal, good, excellent}

Figure 4. Structure of the canonical context model

The tailoring mechanism elements like tailoring rules and its function are better to understand in the next section through an example.

V. SOFTPROF-UP IN ACTION

This section aims to show how the different SoftProF-UP elements are used and interact to generate adapted processes. It will be shown in a particular way how to adapt the process of the implementation discipline so that it meets the particular project characteristics.

A. Process model of implementation discipline

The process model that will be adapted in this section was illustrated in Figure 3, where its implementation in caSPEMTool in Figure 5.

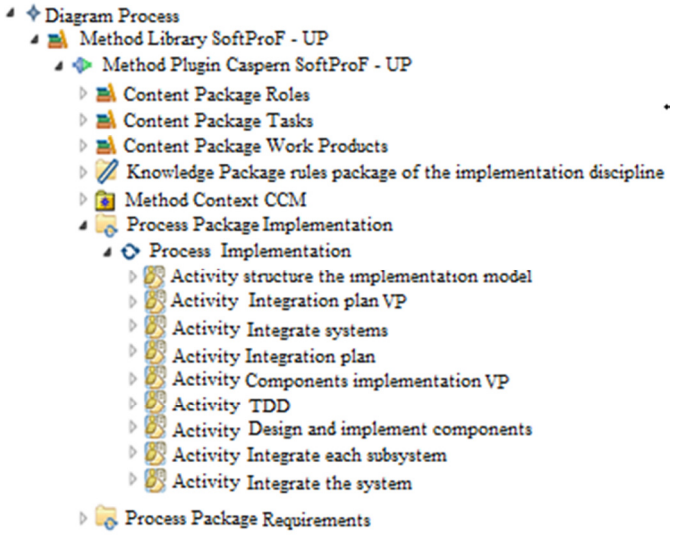

Figure 5. Implementation Process defined in caSPEMTool.

A. CCM Configuration

To adapt the software process, the CCM defined in Figure 4 must be instantiated or configured. It is important to remember that the CCM initially formulated in this work was achieved from the contextual information found in the literature on adaptation elements used explicitly in the Unified Process adaptation, therefore the CCM serves as a starting point to add elements of the context that are necessary depending on the needs of those responsible for adaptation. A CCM configuration consists of determining which dimensions, context attributes, and their corresponding values are suitable for modeling the project particular context. For this case, we will illustrate a project context defined by the context attributes and attribute values as shown in Table 1.

Figure 6 shows a part of CCM implementation and a configuration specific using the caSPEMTool tool. The context model configuration is shown in the lower part of the figure, which is made up of the four context attributes defined in Table 1, and it is particularly shown that the context attribute called type project of the configuration has value maintenance; similarly, the other context attributes of the configuration comply with the values defined in Table 1.

Context attribute	***Value***
Project type	Maintenance
Project size	Small
Team size	Small
Technical Experience	Good

Table 1. Context Attribute for CCM configuration

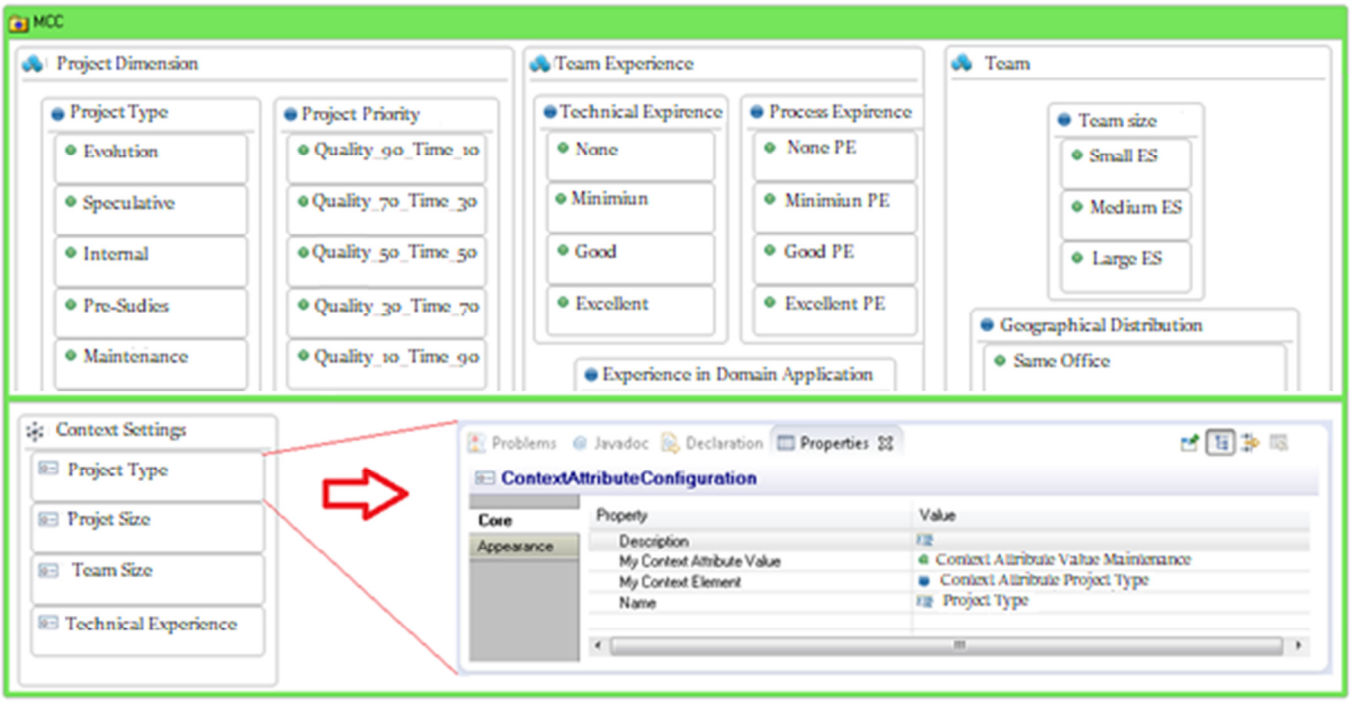

Figure 6. MCC and Configuration

A. Scope Definition

To establish the scope, it is necessary to look for the relationships between the context attributes and the process variation points. In Table 2, the implementation scope discipline process is shown through the relationships between the context elements and the process variability, in the first table column the context attributes are defined, which are involved in the CCM configuration, and the following columns refer to the process features of the implementation discipline.

The scope table allows establishing under which values of the configuration attributes of context the process elements should be selected. For the particular case, Table 2 presents the attributes values of the CCM configuration, according to Table 1, and the relationship with the process variability. The relationships are established with truth values, which indicate whether the activity is part of the adapted process or not, and also if it is the case the name of an activity that will take the place of the variation point.

A. Tailoring rules

The Figure 7, 8 and 9 define the decision trees that guide the implementation of the process rules of the implementation discipline, taking into account the scope table defined this discipline. The green nodes in the trees indicate that if their evaluation is true the variation point will be part of the adapted process, otherwise it will not be part of the process, yellow nodes.

The first two conditional nodes, in Figure 7, indicate that the "Integration Plan" optional variation point, if the context configuration takes any of the two values, will be replaced by the "Integration Plan" activity. The third conditional node indicates that if its evaluation is true, the variation point will be replaced by the "Integrate Systems" activity. The last conditional defines the situations where the variation point will not be part of the adapted process. In the same way, trees are defined as tailoring rules for the variation point of "Design and Components Implementation " and "Integrate each Subsystem" as shown in the figures, Figure 8 and Figure 9.

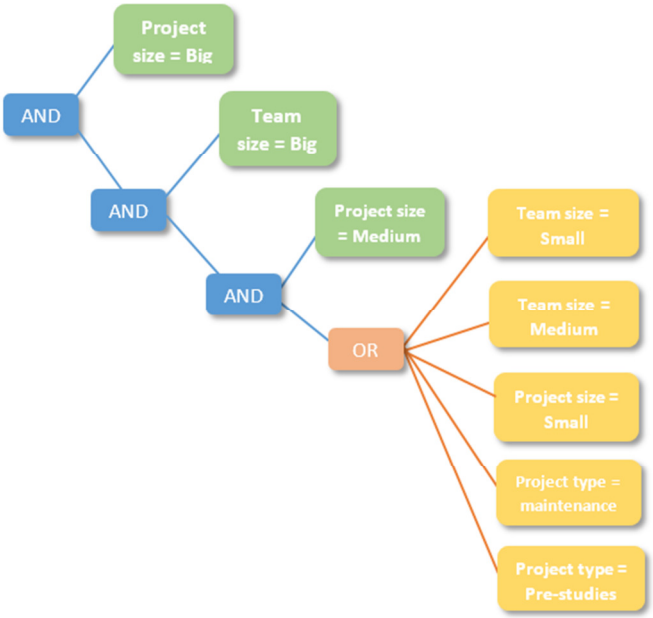

Figure 7. Tailoring rule like decision tree for the planning integration feature

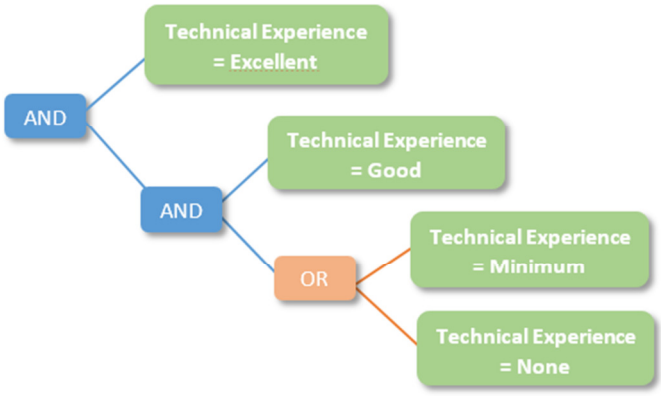

Figure 8. Tailoring rule like decision tree for design and components implementation

Context Attributes	Process Features	Integration Plan			Design and Components Implementation	Systems Integrate
Project Type	Pre-studies	None	FALSE	None		
	Maintenance	None	FALSE	None		
Project Size	Small	None	FALSE	None		FALSE
	Medium		TRUE	Integrate Systems		TRUE
	Big	Integration plan	TRUE			TRUE
Team Size	Small		FALSE			FALSE
	Medium		FALSE			
	Big	Integration plan	TRUE			
Technical Experience	None		---		Components Implementation	
	Minimum		---		Components Implementation	
	Good		---		TDD	
	Excellent		---		TDD	

Table 2. Scope of the SoftProf-UP Implementation discipline process

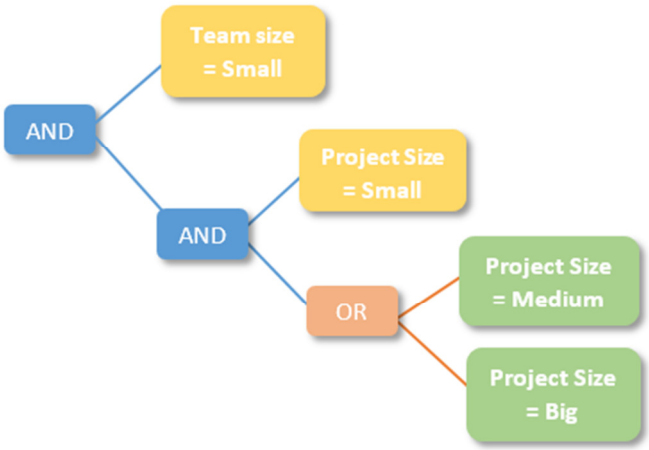

Figure. 9. Tailoring rule like decision tree for Integrate Each Subsystem Feature.

B. Implementing and executing tailoring rules

The tailoring rules goal to transform the general process into a specific process taking into account the configuration of the project context model. The rules are composed of disjunction and conjunction conditional operators that compare the desired context with the specific context of each project and that trigger adaptation actions on process elements. For the tailoring rules implementation, during the modeling of the implementation discipline in caSPEMTool, the rules defined as decision trees of the previous section were taken into account. The first rule implementation is defined in Figure 10, where the first and second conditional of the rule allows defining under what circumstances the optional variation point "Integration Plan" is replaced by the activity "Integration Plan" by means of the action shared by the two conditionals. Similarly, the third conditional allows establishing when the optional variation point is replaced by the "Integrate Systems" activity. In the fourth conditional case, if the context configuration takes a value of the five possible ones to evaluate, the optional variation point will be eliminated. A characteristic of caSPEMTool is that in the Rule Context definition a context attribute can take several values as shown in the last conditional of Figure 9, where the context attribute Team Size can take the values of Medium TE or Small TE and in the same way the context attribute Project Type can take the values of Maintenance or Pre-Studies. Therefore, in the last conditional implementation, only three Rules Context are defined for the five possible values of the context attributes.

The tailoring rule for the "Design and Components Implementation" variation point is defined in Figure 11, Where first conditional, left side of the figure, corresponds

under which values of the context model configuration, the variation point will take the activity "Components Implementation " and the second conditional defines when TDD will take the place of the variation point

The tailoring rule for the characteristic "Integrate each Subsystem", Figure 12, defines only when the activity must be eliminated, due to it is not necessary to define for which values of the context model configuration the activity is part of the process, this is because in caSPEMTool, if the activities are not eliminated or updated by means of the variability defined in SEPM2.0, the process does not change and they will always be part of the process.

The result of adapting the implementation process with the previously defined rules, taking into account the project configuration in Table 1, as shown in Figure 13.

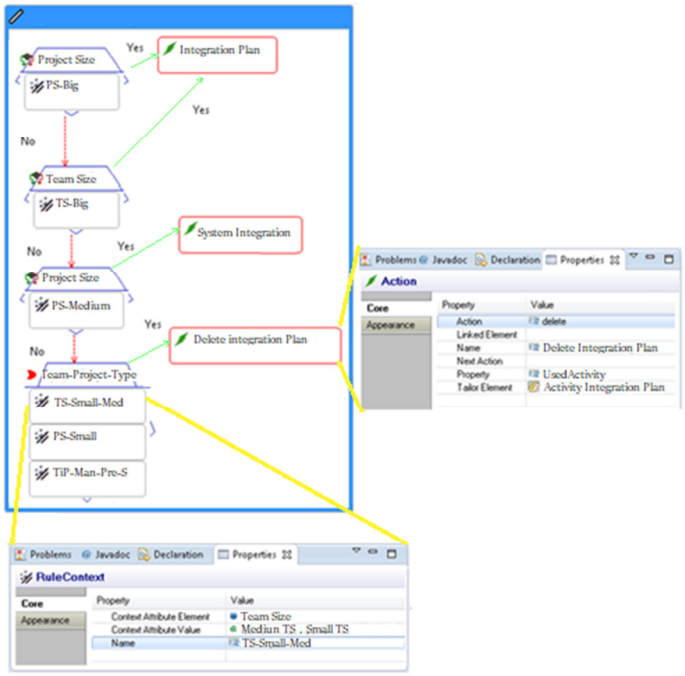

Figure 10. Tailoring rule implementation for planning integration feature

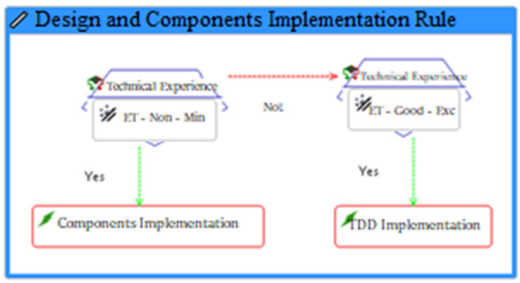

Figure 11. Tailoring rule implementation for Design and Components Implementation

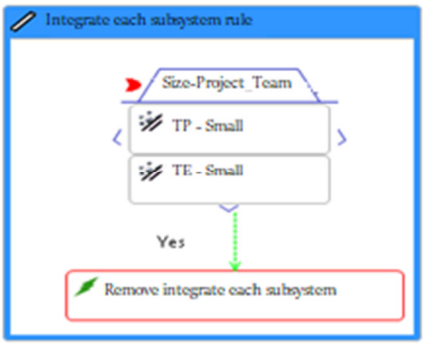

Figure 12. Tailoring rule implementation for Integrate each Subsystem

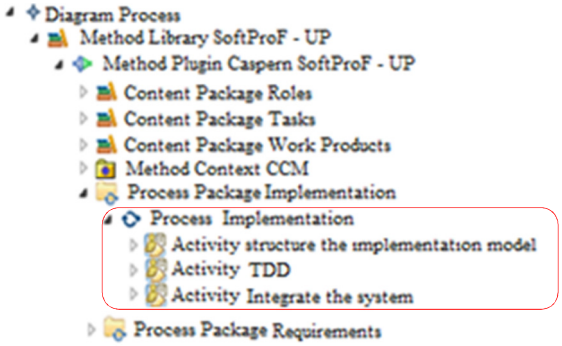

Figure 13. implementation Process Adapted

VI. CASE STUDY

To determine the applicability of the canonical family proposed in this work, the case study development of the holistic type (the case as a whole) and descriptive was proposed. This case study focuses on determining the usefulness of the requirements process of the canonical family, as a starting point, in the family definition their own of an SME software developer. The steps followed to define the case study according to Runeson and Höst [27] are described below

A. Case Study Design

- Research question: The question is: Is canonical SoftProF-UP a useful basis, as a starting point, for the processes family in a small organization?
- Case Study Objective: Determine the canonical basis applicability of SoftProF-UP in one a small software development company.
- Case Study Selection: this case study is holistic and descriptive, with a unit of analysis, which corresponds to the software process of a developer company. The organization was selected under the criterion that it was a small organization and in addition, its availability for this work was considered, in this case, the company Rhiscom, a small Chilean software organization.

- Case Context: Rhiscom is a small Chilean software company in the Latin American market, located in the city of Santiago, Chile, whose software process has been inspired by the Unified Process.
- Measurements and instruments design of the case study: The metrics and instruments that were used for this case study are shown in Table 3.

B. Case Study Development

The company initially had a documented software process, however distant from the practice and without sufficient formalization to take advantage of the systematic adaptation approach. Therefore, the case study was developed in three stages to be able to achieve the use of the systematic approach to process adaptation. *a) Stage Formalization of the current process (practiced) Rhiscom.* In the formalization, the Rhiscom process two months were spent, for which it was necessary to know the company scope and the process followed in the development of its products. In the definition and formalization process, we asked about the company scope and the process followed for the development of the product by consulting documents and process descriptions. The formalization was done through meetings to elicit each process workflow including roles, activities and work products involved in each process area. These sessions result was the process documentation and specification applied in each area in the EPFC tool. b) Family analysis and design stage. For this case study stage, a comparison and homologation were made between each one of the activities defined in Canonical SoftProF-UP and those corresponding to the Rhiscom requirements process. In order to develop this stage, the formalized process, the documented process and the canonical processes family were taken as inputs. To make the analysis and design of the Rhiscom - SoftProF- UP process a comparison was made for each process element, the comparison was initiated hierarchically by the activities. In this comparison, each activity was taken and its compatibility between the two processes was verified in such a way that activities capable of homologation were identified and then focused on the comparison of artifacts and roles. As part of this stage, it was necessary to carry out the elements elimination and addition in SoftProF-UP in a way that fulfilled the requirements of the Rhiscom requirements' own specification. Table 4 shows the total elements of each process and Table 5 shows the comparable elements total to define the Rhiscom process through the Canonical process family. c) The resulting process formalization as a processes family: The process formalization was carried out taking into account the previous stage results where caSPEMTool was used as a tool to define the process elements, context and tailoring rules that are part of Rhiscom- SoftProF-UP. First, the context model was:

Indicators	Metrics	Instruments
Activities utility AU: Family activities percentage, useful in the company process definition. ***AU= (*SANCompany / ATNDS*)*100***	SoftProF-UP activities number used in the company (**SAN*Company*)** Activities total number defined in SoftProF-UP (**ATNDS**)	Rhiscom process model SoftProF-UP process model
Activities coverage AC: Tasks percentage of the company that is covered by SoftProF-UP ***AC = (CANCCompany / ATNCompany) * 100***	Company activities number covered by SoftProF-UP (***CANCCompany***) Activities total number of the company (***ATNCompany***)	Rhiscom process model SoftProF-UP process model
Roles utility RU: Family roles percentage, useful in the company process definition. ***RU= (*SRNCompany / RTNDS*)*100***	SoftProF-UP Roles Number used in the company (**SRNCompany**) Roles total number defined in SoftProF-UP (**RTNDS**)	Rhiscom process model SoftProF-UP process model
Roles coverage RC: Roles percentage of the company that is covered by SoftProF-UP **RC = (CRNCCompany / RTNCompany) * 100**	Company Roles number covered by SoftProF-UP **(CRNCCompany)** Roles total number of the company (**RTNCompany**)	Rhiscom process model SoftProF-UP process model
Work products utility **WPU:** Family work products percentage, useful in the company process definition. ***WPU= (*SWPCompany/ WPTNS*)*100***	SoftProF-UP work products number used in the company (**SWPCompany**) Work products total number defined in SoftProF-UP (**WPTNS**)	Rhiscom process model SoftProF-UP process model
Work products coverage WPC: Work products percentage of the company that is covered by SoftProF-UP. **WPC=(CWPNCCompany/** WPTNCompany)***100**	Company work products number covered by SoftProF-UP **(CWPNCCompany)** Work products total number of the company (**WPTNCompany**)	Rhiscom process model SoftProF-UP process model
Rules utility RUU: Family rules percentage, useful in the company process definition. **RUU=(SRNUCompany/ RTNBS)**	SoftProF-UP rules number used in the company (**SRNUCompany**) Rules total number based on SoftProF-UP (**RTNBS**)	Rhiscom process model SoftProF-UP process model
Rules coverage RUC: Rules percentage of the company that is covered by SoftProF-UP. **RUC = (CRNCCompany/ RTNCompany) * 100**	Company rules number covered by SoftProF-UP **(CRNCCompany)** Rules total number of the company (**RTNCompany**)	Rhiscom process model SoftProF-UP process model

Table 3. Measurements and instruments design of the case study

formalized taking into account the dimensions, attributes, and context appropriate values for the company. Second, the process structure was formalized, starting with phases, disciplines, activities, work products, and roles through caSPEM 2.0. In the formalization some tests were made to verify that the family under specific considerations worked correctly, the variation points and optionality defined in Rhiscom-SoftProF-UP were tested. As a final result, a

formalized process model was obtained in caSPEM 2.0 according to SPEM 2.0

Processes	Activities	Work products	Roles	Rules
SoftProF-UP	9	10	3	2
Rhiscom	10	6	6	1

Table 4. Total elements of the requirements processes of canonical SoftProF-UP and Rhiscom

Homologation	Activities	Work products	Roles	Rules
Total	6	5	2	1

Table 5. Requirements process elements homologated between SoftProF-UP and Rhiscom

C. Case Study Results

To make the comparison between the processes it was necessary to determine the elements amount that made them, activities, work products, roles, optionality and variation points, to later make an adequate comparison. It was found that a activities number defined in the two processes is quite similar. The difference is more notable among the roles and products of work because the roles defined in Rhiscom are twice those of SoftProF- UP and the work products number defined in SoftProF- UP is greater, almost double, of those defined in Rhiscom. The activities percentage defined in SoftProF-UP, which corresponded to the company's process model is 66%. The coverage of SoftProF-UP activities with respect to the Rhiscom process activities was 60%. View table 6.

The roles usefulness is similar to that of the activities, the roles utility percentage is 66%. The roles coverage is 33%, where the company process model defines four different roles to those of SoftProF-UP, and SoftProF-UP considers a totally different role. View table 7.

The work products utility percentage is of 50%, even though SoftProF-UP defines the double Rhiscom work products. Five of the ten SoftProF-UP work products are defined similarly to those of the company. The correspondence percentage is 83% taking into account that only a company work product is not defined in the canonical family. View table 8.

The rules utility is 50%, although the rule scheme is more useful than the implementation logic. The rule's coverage is 100% because the company only defined a tailoring rule which was defined structurally similar to that of SoftProF-UP. Although the coverage is high, this does not mean that the rules definition in organizations can be completely inherited from the canonical family, knowing that the companies' processes application areas are different, the rules elaboration is a recurrent work that each organization must perform, so it is necessary to define tailoring patterns that help organizations in the tailoring rules definition appropriate to their process and organizational context.

Activities	Activities number	Percentage
Activities utility of SoftProF-UP	6	66%
SoftProF-UP activities coverage above the company activities	6	60%

Table 6. Utility and Coverage of SoftProF-UP activities

Roles	Roles number	Percentage
Roles utility of SoftProF-UP	2	66%
SoftProF-UP roles coverage above the company roles	2	33%

Table 7. Utility and Coverage of SoftProF-UP roles

Work products	Work products number	Percentage
SoftProF-UP work products used in the company	5	50%
Company work products found in SoftProF-UP	5	83%

Table 9. Utility and Coverage of SoftProF-UP work products

Rules	Rules number	Percentage
Rules utility of SoftProF-UP	1	50%
Company rules found in SoftProF-UP	1	100%

Table 10. Utility and Coverage of SoftProF-UP tailoring rules

D. Results Analysis

The canonical model correspondence with the company process model is 66%, a percentage that indicates that more than half activities served as the basis for the Rhiscom-SoftProF-UP process definition. The SoftProF-UP activities coverage with respect to the Rhiscom process activities by 60% shows that a good percentage of the canonical family activities covered the Rhiscom-SoftProF-UP process definition. The roles utility can be considered as good so the difference between the processes roles number is double, since it is 66%. The roles coverage reaches 33%, which means that there is an overflow between the process families scope at the role level. These results obtained in this case study are revealing and allow inferring that canonical SoftProF-UP serves as a basis for defining, in the company context, the software process as a process family with some limitations not minor. Particularly the activities and roles usefulness percentages is greater than 65%, of the work products is 50% and taking into account that they are the basic elements of the process definition, they reinforce that SoftProF-UP has an applicability high, as a basis for defining families, which depends on both the company formalized process and its business scope. SoftProF-UP seeks to provide the elements and the necessary infrastructure that serves as an initial point to generate processes families and not a compatibility with all possible UP-based software processes. The difference degree between the processes defined in a company and SoftProF-UP can be considered to be partly due to the compliance degree that business processes have with UP, taking into account that the SoftProF-UP definition starts from the same UP and some of its variants.

In addition, the case study allows us to see that canonical SoftProF-UP is small to be followed and manipulated by small software companies. The canonical solution limitation is given by the limitations in the tailoring rules collected from the literature with the generalization purpose. At a particular level, the rules are realizable, but at a general level, context elements and decision are missing to solve the variable elements according to the contexts. This resulted, in the extraction of three tailoring patterns that are provided as support to companies for the particular definition of their software processes family.

VII. CONCLUSIONES

This work shows a canonical processes family, based on the unified process, whose purpose is to serve as a base for the software companies to define their own process family, and to take advantage of an innovation high-level adaptation mechanism using an MDE strategy. SoftProF-UP seeks to provide the necessary elements and infrastructure to serve as a starting point in the processes families creation and not an accuracy with all the possible software processes that are in UP, which is a titanic task and of little use. The idea of this work was to determine a process family based on the unified process, under the hypothesis that, the amplitude of its use and its adaptation to multiple contexts, would give enough empirical evidence for its formulation. However, the scientific and industry reports under the study conducted in this work did not provide sufficient evidence. From this, the conclusion was reached that the Unified Process is a broad-spectrum process framework, which limits its formulation as a process family, given that a family requires action domain, as well as the scope. Although the empirical evidence was insufficient to formulate a large process family, it did serve as a basis for determining that each business, in a determined action domain and therefore with a smaller scope, as can be evaluated in the case study, it is possible to define a process family at the organizational level based on the Unified Process assets. Given the previous conclusion, this work focuses on strengthening the process family construction at the organizational level since it is not feasible for each organization to have to do it from the beginning. In addition, this work proposes a canonical process family established with scarce empirical evidence and with some typical examples, which facilitate organizations that have the Unified Process as a reference, to define their own process family and thus achieve a cost-effective adaptation of their process. This allows them to use an appropriate process for each software project they face.

The case study consisted in determining the SoftProF-UP canonical applicability base in Rhiscom, a small Chilean software development company, in which it was possible to evaluate that SoftProF-UP has a high applicability and some limitations in the formulation of adaptation rules, considering both the company formalized process and its business scope.

The process family resulting from this work is a significant advance and initial base that must still be refined in its components through empirical evidence and research reports. The canonical family based on the Unified Process can be evolved by the software industry in such a way that this starting point reduces the effort in the organizations for its reuse.

ACKNOWLEDGEMENT

This work was partially funded by the project "Red de formación de talento humano para la innovación social y productiva en el departamento del Cauca -INNOVACCIÓN CAUCA" executed by the Universidad of Cauca under code 3848 and by project 4519 of the investigations directorate of the Universidad of Cauca.

REFERENCES

[1] W. S. Humphrey, «Managing the Software Process,» *Addison-Wesley Longman,* 1989.

[2] I. Yoon, S. Min y D. Bae, «Tailoring and Verifying software Process,» *Eighth Asia-Pacific Software Engineering Conference (APSEC'01).,* p. 202, 2001.

[3] G. Hanssen, F. Bjørnson y H. Westerheim, «Tailoring and Introduction of the Rational Unified Process.,» *P. Abrahamsson et al. (Eds.): EuroSPI 2007, LNCS 4764. Springer-Verlag Berlin Heidelberg,* p. 7–18, 2007.

[4] G. K. Hanssen, H. Westerheim y F. O. Bjørnson, «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, 2005.

[5] I. Jacobson, G. Booch y J. Rumbaugh, «Unified Software Development Process,» 1999.

[6] J. Hartmann, L. Fontoura y R. Price, «Using Risk Analysis and Patterns to Tailor Software Processes,» *Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS),* 2005.

[7] J. Hurtado, A. Quispe, M. Bastarrica y S. Ochoa, «A MDE Production Strategy for Software Process Lines,» *International Conference on Software and Systems Processes,* 2011.

[8] H. Washizaki, «Building Software Process Line Architectures from Bottom Up,» *Product-Focused Software Process Improvement,* pp. 415-421, 2006.

[9] E. Pereira, R. Bastos y T. Oliveira, «A Systematic Approach to Process Tailoring,» *International Conference on Systems Engineering and Modeling. ICSEM,* 2007.

[10] C. Coelho, «Modelo de Adaptación de proceso software,» *Master' Thesis. Federal University of Pemambuo – UFPE,* 2003.

[11] C. Patel, S. Cesare, N. Iacovell y A. Merico, «A Framework for Method Tailoring: A Case Study,» *Proc. 2nd OOPSLA Workshop on Method Engineering for Object-Oriented and Component-Based Development,* 2004.

[12] P. H. Ruiz, C. E. Maya, W. L. Pantoja y J. A. Hurtado, «Guía de Adaptación del Proceso Unificado a Proyectos Específicos para las Pymes Desarrolladoras de Software,» *Cogreso Colombiano de Computación ,* 2010.

[13] OMG, «Software & Systems Process Engineering Metamodel (SPEM),» 2007.

[14] AUP, «Agile UP (AUP),» 2005. [En línea]. Available: http://www.ambysoft.com/unifiedprocess/agileUP.html. [Último acceso: 06 Abril 2011].

[15] R. Balduino, «Basic Unified Process: A Process for Small and Agile,» *IBM*.

[16] OpenUP, «Open Unified Process (OpenUP) y Open Unified Process Basic (OpenUP/Basic),» 2007. [En línea]. Available: http://epf.eclipse.org/wikis/openup/. [Último acceso: 06 Abril 2011].

[17] EUP, 2000. [En línea]. Available: http://www.enterpriseunifiedprocess.com/. [Último acceso: 06 Abril 2011].

[18] B. Simidchieva, L. Clarke y L. Osterweil, «Representing Process Variation with a Process Family,» *In qing Wang, Dietmar Pfahl, and David M. Raffo, editors, ICSP´2007, volumen 4470 of LNCS,* pp. 109-120, 2007.

[19] A. Cass, A. Lerner y E. McCall, «Little JIL/Juliette: A Process Definition Language and Interpreter,» *Proceedings of the 2000 International Conference on Software Engineering,* pp. 754 - 757, 2000.

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

[21] R. France y B. Rumpe, «Model-driven Development of Complex Software: A Research Roadmap,» *FOSE '07 2007 Future of Software Engineering,* pp. 37-54, 2007.

[22] J. Hurtado y C. Bastarrica, «Process Model Tailoring as a Means for Process Knowledge Reuse,» *In Proceedings of the 2nd Workshop on Knowledge Reuse. Falls Church Virginia, USA,* 2009.

[23] J. A. Hurtado y M. C. Bastarrica, «Building Software Process Lines with CASPER,» *International Conference on Software and Systems Processes. ICSSP,* 2012.

[24] B. Kitchenham, «Procedures for performing systematic reviews,» *Joint Technical Report,* 2004.

[25] S. Ochoa, A. Quispe, J. Hurtado y C. Bastarrica, «Project Context Definition Method,» *Tech. rep., Computer Science Department, Universidad de Chile,* 2012.

[26] P. H. Ruiz, J. A. Hurtado y M. C. Camacho, «Analizando la Variabilidad Del Proceso Unificado con la Tècnica de Washizaki,» *Gestión de la Técnologia Informática,* 2013.

[27] P. Runeson y M. Höst , «Guidelines for conducting and reporting case study research in software engineering.,» *(D. Sjoberg, Ed.) Empirical Software Engineering,* pp. 131-164, 2009.